UNIVERSIDADE FEDERAL DE GOIÁS
INSTITUTO DE INFORMÁTICA

MÁRCIO DIAS DE LIMA

# Mínimos quadrados para problemas de múltiplas classes envolvendo twin support vector machine e aplicações de mineração de dados

Goiânia
2018

sistema de bibliotecas ufg

PRPG
PRÓ-REITORIA DE PÓS-GRADUAÇÃO

UFG

**TERMO DE CIÊNCIA E DE AUTORIZAÇÃO PARA DISPONIBILIZAR VERSÕES ELETRÔNICAS DE TESES E DISSERTAÇÕES NA BIBLIOTECA DIGITAL DA UFG**

Na qualidade de titular dos direitos de autor, autorizo a Universidade Federal de Goiás (UFG) a disponibilizar, gratuitamente, por meio da Biblioteca Digital de Teses e Dissertações (BDTD/UFG), regulamentada pela Resolução CEPEC nº 832/2007, sem ressarcimento dos direitos autorais, de acordo com a Lei nº 9610/98, o documento conforme permissões assinaladas abaixo, para fins de leitura, impressão e/ou *download*, a título de divulgação da produção científica brasileira, a partir desta data.

**1. Identificação do material bibliográfico:** **[ ] Dissertação** **[ X ] Tese**

**2. Identificação da Tese ou Dissertação:**

Nome completo do autor: Márcio Dias de Lima

Título do trabalho: Mínimos quadrados para problemas de múltiplas classes envolvendo twin support vector machine e aplicações de mineração de dados

**3. Informações de acesso ao documento:**

Concorda com a liberação total do documento [ ] SIM [ X ] NÃO[1]

Havendo concordância com a disponibilização eletrônica, torna-se imprescindível o envio do(s) arquivo(s) em formato digital PDF da tese ou dissertação.

Márcio Dias de Lima
Assinatura do autor[2]

Ciente e de acordo:

Assinatura do orientador[2]

Data: 07 / 12 / 2018

[1] Neste caso o documento será embargado por até um ano a partir da data de defesa. A extensão deste prazo suscita justificativa junto à coordenação do curso. Os dados do documento não serão disponibilizados durante o período de embargo.

Casos de embargo:
- Solicitação de registro de patente
- Submissão de artigo em revista científica
- Publicação como capítulo de livro
- Publicação da dissertação/tese em livro

[2] A assinatura deve ser escaneada.

MÁRCIO DIAS DE LIMA

# Mínimos quadrados para problemas de múltiplas classes envolvendo twin support vector machine e aplicações de mineração de dados

Tese apresentada ao Programa de Pós–Graduação do Instituto de Informática da Universidade Federal de Goiás, como requisito parcial para obtenção do título de Doutor em Ciência da Computação.

**Área de concentração:** Ciência da Computação.

**Orientador:** Prof. Dr. Rommel Melgaço Barbosa

Goiânia
2018

Ficha de identificação da obra elaborada pelo autor, através do
Programa de Geração Automática do Sistema de Bibliotecas da UFG.

Dias de Lima, Márcio
Mínimos quadrados para problemas de múltiplas classes envolvendo twin support vector machine e aplicações de mineração de dados [manuscrito] / Márcio Dias de Lima. - 2018.
CXXVII, 127 f.

Orientador: Prof. Rommel Melgaço Barbosa.
Tese (Doutorado) - Universidade Federal de Goiás, Instituto de Informática (INF), Programa de Pós-Graduação em Ciência da Computação em rede (UFG/UFMS), Goiânia, 2018.
Bibliografia.
Inclui siglas, símbolos, lista de figuras, lista de tabelas.

1. Mineração de dados. 2. SVM. 3. TWSVM. 4. Mínimos Quadrados. 5. Classificação. I. Melgaço Barbosa, Rommel, orient. II. Título.

CDU 004

UNIVERSIDADE FEDERAL DE MATO GROSSO DO SUL
UNIVERSIDADE FEDERAL DE GOIÁS
INSTITUTO DE INFORMÁTICA
DOUTORADO EM CIÊNCIA DA COMPUTAÇÃO

# Ata de Defesa de Tese de Doutorado

Aos sete dias do mês de dezembro de dois mil e dezoito, no horário das catorze horas, foi realizada, nas dependências do Instituto de Informática da UFG, a defesa pública da Tese de Doutorado do aluno Márcio Dias de Lima, matrícula no. 2015 0211, intitulada **"Mínimos quadrados para problemas de múltiplas classes envolvendo Twin Support Vector Machine e aplicações de mineração de dados"**.

A Banca Examinadora, constituída pelos professores:

Prof. Dr. Rommel Melgaço Barbosa – INF/UFG - orientador

Prof. Dr. Ronaldo Martins da Costa – INF/UFG

Prof. Dr. Thierson Couto Rosa – INF/UFG

Prof. Dr. Helton Saulo Bezerra dos Santos – UnB

Profa. Dra. Katia Kelvis Cassiano Lozano – FIC/UFG

emitiu o resultado:

(X) Aprovado

( ) Aprovado com revisão

(A Banca Examinadora deve definir as exigências a serem cumpridas pelo aluno na revisão, ficando o orientador responsável pela verificação do cumprimento das mesmas.)

( ) Reprovado

com o seguinte parecer: _____

_____

_____

_____

Prof. Dr. Rommel Melgaço Barbosa

Prof. Dr. Ronaldo Martins da Costa

Prof. Dr. Thierson Couto Rosa

Prof. Dr. Helton Saulo Bezerra dos Santos

Profa. Dra. Katia Kelvis Cassiano Lozano

Todos os direitos reservados. É proibida a reprodução total ou parcial do trabalho sem autorização da universidade, do autor e do orientador(a).

**Márcio Dias de Lima**

Possui graduação em Licenciatura em Matemática pela Universidade Federal de Goiás - UFG (2001), especialização em matemática pela UFG (2004) e mestrado em matemática pela UFG (2009), com ênfase em Álgebra. Atualmente é professor no Instituto Federal de Educação, Ciência e Tecnologia de Goiás, atuando no câmpus Goiânia e tem interesses em pesquisas nas áreas de mineração de dados, complexidade de algoritmos, otimização e álgebra.

Dedico esta tese a minha esposa Juliana Lima, você me faz melhor a cada dia, aos meus filhos Gustavo Lima e Marina Lima, a minha mãezinha Wilma e ao meu querido e amado pai, José Francisco de Lima *"in memorian"*.

# Agradecimentos

Na realização desta tese, contei com o apoio direto e indireto de várias pessoas e instituições às quais sou profundamente grato.

Ao meu orientador, professor Dr. Rommel Melgaço Barbosa, pela orientação prestada, pelo seu incentivo, disponibilidade e apoio que sempre demonstrou. Aqui lhe exprimo a minha gratidão.

A todos os amigos que colaboraram e auxiliaram na elaboração do presente estudo, pela paciência, conversas, brincadeiras: Weder, Camila, Thiago, Daniel, Jailton, Audir, Leonardo, e de modo especial a Nattane e Diego pelas parcerias e amizade, que muito contribuíram no desenvolvimento deste trabalho.

Não poderia deixar de agradecer à minha família por todo o apoio, pela força e pelo carinho que sempre prestaram ao longo de toda a minha vida acadêmica, principalmente minha mãe, que sempre me incentivou, torceu e sofreu comigo todos os momentos desta etapa. Infelizmente meu avô partiu em 2009 e não está presente neste momento tão importante da minha vida, mas tenho certeza que iria se orgulhar muito. Ao longo do doutorado, perdi duas pessoas muito importantes em minha vida e formação, minha avó (Maria dos Santos Dias) e meu pai (José Francisco de Lima), e é com muito alegria que dedico essa vitória a vocês, que fazem muita falta, e que, com certeza, iriam comemorar mais uma etapa concluída em minha vida.

Ao Instituto Federal de Goiás, pela concessão do afastamento, o que permitiu que eu me dedicasse integralmente ao doutorado, e agradeço também a todos os amigos que lá atuam.

Aos servidores técnicos administrativos do Instituto de Informática da Universidade Federal de Goiás, Mariana, Mirian e Patrícia, pela atenção e dedicação em seus trabalhos realizados para o bom andamento do doutorado.

Agradeço à banca examinadora, Prof. Dr. Helton Saulo Bezerra dos Santos, Prof$^{a}$. Dra. Kátia Kelvis Cassiano Lozano, Prof. Dr. Ronaldo Martins da Costa e ao Prof. Dr. Thierson Couto Rosa que me deram consideráveis sugestões e contribuições para o texto final.

Muito obrigado aos amigos, Daniela Dering e ao Prof. Dr. Jair Alcindo de Lobo Melo pelas correções na escrita deste estudo.

Agradeço aos meus filhos Gustavo e Marina, que mesmo tentando estar o mais presente no dia a dia de vocês, devo ter falhado em muitos momentos, mas vocês sempre estavam ali ao meu lado, me incentivando e motivando a continuar. Obrigado por vocês serem estes filhos maravilhosos, que tenho muito orgulho de tê-los.

E a você minha esposa Juliana Lima, por estar ao meu lado mesmo com todo o meu mau humor, estresse e minha falta de paciência em todo este período. Quero te agradecer imensamente por ser esta pessoa maravilhosa, que me ajudou, ajuda e ajudará em todos os momentos, por apresentar sempre um sorriso, quando sacrificava os dias, as noites, os fins-de-semana e os feriados em prol da realização deste estudo. Por tudo isso, muito obrigado.

Enfim, quero demonstrar o meu agradecimento, a todos aqueles que, de um modo ou de outro, tornaram possível a realização da presente tese. A todos, o meu sincero e profundo muito obrigado!

"O sucesso é a soma de pequenos esforços repetidos dia após dia."

**Robert Collier.**

.

# Resumo


Lima, Márcio Dias de. **Mínimos quadrados para problemas de múltiplas classes envolvendo twin support vector machine e aplicações de mineração de dados**. Goiânia, 2018. 127p. Tese de Doutorado. Instituto de Informática, Universidade Federal de Goiás.

Mineração de dados é uma área emergente, devido à quantidade crescente de dados produzidos nas mais variadas áreas. Neste contexto, as máquinas de vetores de suportes duplos (TWSVM) têm recebido muita atenção por partes de pesquisadores. Elaboramos um algoritmo seletor de variáveis e um algoritmo para problemas de múltiplas classes baseado no TWSVM. Esse algoritmo de aprendizagem com saídas ternárias $\{-1,0,+1\}$ é baseado na teoria de vetores de suporte de Vapnik, e avalia todas as amostras de treinamento com uma estrutura $1-\times-1-\times-restante$, durante a fase de decomposição. Uma das principais vantagens do algoritmo proposto é que utilizamos a versão mínimos quadrados para problemas de múltiplas classes, onde é necessário resolver dois sistemas de equações lineares, ao contrário do TWSVM em que é necessário resolver dois problemas de programação quadrática. Implementamos o princípio de minimização do risco estrutural, a fim de melhorar a habilidade de generalização. Utilizamos a fórmula de Sherman-Morisson-Woodbury com o intuito de reduzir a complexidade na formulação não linear do algoritmo. Abordamos também aplicações de técnicas de mineração de dados que combinam a utilização de técnicas analíticas com algoritmos de mineração de dados na classificação de diversas amostras, podendo ser considerada uma excelente ferramenta na detecção de fraudes de diversos tipos, como na verificação de produtos cultivados nos sistemas orgânicos ou convencional, a fim de diferenciá-los, e também na verificação quanto a origem geográfica de vinhos produzidos de um determinado tipo de uva.



**Palavras–chave**

Mineração de dados, SVM, TWSVM, Mínimos Quadrados, Classificação.

# Abstract

Lima, Márcio Dias de. **Least squares for multiple class problems involving twin support vector machine and data mining applications**. Goiânia, 2018. 127p. PhD. Thesis. Instituto de Informática, Universidade Federal de Goiás.

Data mining is an emerging area due to the increasing amount of data available in a variety of fields. In this context twin support vector machine (TWSVM) has attracted the attention of several researchers. In this thesis, we developed a feature selector algorithm and an algorithm for multi-class problems based on TWSVM. This learning algorithm with ternary outputs $\{-1, 0, +1\}$ is based on the Vapnik support vector theory, and evaluates all training samples with a $1 - \times - 1 - \times -$ *rest* structure during the decomposition phase. One of the main advantages of the proposed algorithm is the use of the least squares version for multi-class problems, where it is necessary to solve two systems of linear equations instead of two quadratic programming problems in TWSVM. We also implemented the principle of minimization of structural risk in order to improve the generalizability. The Sherman-Morisson-Woodbury formula is applied to reduce the complexity of the non-linear formulation of the algorithm. We also apply data mining techniques that combine the use of analytical technique with data mining algorithms in the classification of several samples. The developed framework could be an excellent tool for detecting different types of fraud, verifying if products were grown in organic or conventional systems, as well as tracing the region of origin of wine made from a given type of grape.

**Keywords**

Data Mining, SVM, TWSVM, Least Squares, Classification.

# Sumário


Lista de Figuras **15**

Lista de Tabelas **16**

Lista de Siglas **17**

Lista de Notações **19**

1 Apresentação **20**
- 1.1 Motivação 20
- 1.2 Objetivos 22
  - 1.2.1 Objetivo Geral 22
  - 1.2.2 Objetivos Específicos 22
- 1.3 Organização da Tese 22

2 Referencial Teórico **24**
- 2.1 Considerações Iniciais 24
- 2.2 SVM 24
- 2.3 PSVM 25
- 2.4 GEPSVM 25
- 2.5 TWSVM 25
- 2.6 Mínimos quadrados 26
- 2.7 ν-TWSVM 28
- 2.8 Projeção 30
- 2.9 Múltiplas classes 30
- 2.10 Regressão 32
- 2.11 Seletor de variáveis 32
- 2.12 Aplicações 33

3 Máquinas de Vetores de Suporte (SVMs) e extensões **34**
- 3.1 Considerações Iniciais 34
- 3.2 SVM 34
  - 3.2.1 SVM lineares com margens rígidas 38
  - 3.2.2 SVM lineares com margens suaves 42
  - 3.2.3 SVM não lineares 44
  - 3.2.4 Funções *Kernel* 45
- 3.3 Máquinas de Vetores de Suporte Proximal 47
  - 3.3.1 PSVM Linear 47
- 3.4 Máquinas de Vetores de Suporte Proximal por Generalização de Autovalores 48

3.5 Considerações Finais 50

4 Máquinas de Vetores de Suporte Duplo (TWSVM) e extensão **51**
4.1 Considerações Iniciais 51
4.2 TWSVM 51
4.3 TWSVM linear 51
4.4 TWSVM não linear 54
4.5 Extensão baseadas no TWSVM 56
4.5.1 TBSVM 56
4.5.2 Caso Linear do TBSVM 56
4.6 Considerações Finais 59

5 Máquinas de vetores de suportes duplo para problemas de múltiplas classes **60**
5.1 Considerações Iniciais 60
5.2 Estratégias para problemas de múltiplas classes 60
5.3 K-SVCR 62
5.3.1 Twin-KSVC 63
Caso linear Twin-KSVC 63
5.4 LSTKSVC 65
5.5 Considerações Finais 67

6 Melhorias na versão mínimos quadrados para problemas de múltiplas classes do TWSVM **68**
6.1 Considerações Iniciais 68
6.2 Melhorias na versão mínimos quadrados para problemas de múltiplas classes envolvendo twin support vector machine 68
6.2.1 Caso linear 69
6.2.2 Caso não linear 72
6.3 Experimentos Numéricos 77
6.3.1 Seleção de Parâmetros 78
6.3.2 Resultados e discussões 78
6.3.3 Análise estatística 80
6.4 Comentários Finais 83

7 Algumas aplicações **84**
7.1 Considerações Iniciais 84
7.2 Classificação de amostras de laranjas cultivadas nos sistemas orgânico e convencional, utilizando os elementos químicos obtidos pelo ICP-MS 84
7.2.1 Introdução 84
7.2.2 Conjunto de dados analisado 85
7.2.3 Metodologia 85
7.2.4 Resultados 85
7.2.5 Conclusão 87
7.3 Reconhecimento geográfico de vinhos Syrah produzidos na Argentina e Chile 87
7.3.1 Introdução 87
7.3.2 Conjunto de dados analisados 88
7.3.3 Metodologia 88
7.3.4 Resultados 89

7.3.5 Conclusão 91
7.4 Impressão digital e técnicas de aprendizado de máquina para avaliar a autenticidade de vegetais orgânicos e convencionais 91
7.4.1 Introdução 91
7.4.2 Conjunto de dados analisados 92
7.4.3 Metodologia 93
7.4.4 Resultados 93
7.4.5 Conclusão 95
7.5 Classificação em bases de dados médicos, usando um novo algoritmo de seleção de variáveis, combinado com máquinas de vetores de suporte duplo 95
7.5.1 Introdução 95
7.5.2 Seletor de variáveis baseado no twin bounded support vector machine - FSTBSVM 96
7.5.3 Conjuntos de dados analisados 97
7.5.4 Metodologia 98
7.5.5 Resultados 98
7.5.6 Conclusão 102
7.6 Resultados comparativos entre as bases de dados de alimentos e vinhos, utilizando SVM, MLP, RF e FSTBSVM 103
7.7 Considerações Finais 108

**8 Conclusões e Trabalhos Futuros 110**

**Referências Bibliográficas 112**

# Lista de Figuras


3.1 Gráfico com dados fictícios sendo: (a) amostras no plano $\mathbb{R}^2$, (b) alguns possíveis hiperplanos separadores, (c) hiperplano separador com margem máxima, (d) distância entre os hiperplanos de margem máxima. 39

3.2 Gráfico com dados fictícios sendo: (a) com uma amostra entre as margens, mas dentro da superfície de decisão, (b) com uma amostra entre as margens, mas do lado errado da superfície de decisão. 43

3.3 Gráfico das amostras $(-1,-1)$, $(-1,+1)$, $(+1,-1)$ e $(+1,+1)$ com rótulos $\{-1,+1,+1,-1\}$, respectivamente, sendo (a) não linearmente separável, (b) linearmente separável. 45

3.4 Gráfico PSVM linearmente separável. 48

3.5 Gráfico GEPSVM linearmente separável. 50

4.1 Gráfico TWSVM linearmente separável. 54

5.1 Ilustração da estratégia "$1-\times-restante$". 61

6.1 Interpretação geométrica do comportamento do algoritmo. 69

7.1 Gráficos dos rankings utilizando: (a) $\chi^2$ e (b) *RFI*. 89

7.2 Acurácia do SVM, MLP e ELM utilizando o seletor RFI. 90

7.3 Comportamento do classificador SVM em relação às variáveis Ca e Mn, obtidas das amostras de cebola. 93

7.4 Gráfico dos classificadores (SVM, MLP e RF) com variáveis selecionadas pelo $\chi^2$ (a) amostras de alface, (b) amostras de cebola, (c) amostras de pimentão e (d) amostras de tomate. 94

7.5 Fluxograma do nosso FSTBSVM usando a busca em grade para encontrar o modelo otimizado. 98

# Lista de Tabelas


6.1 Bases de dados usados nos experimentos. 78
6.2 Comparação de desempenho dos algoritmos de múltiplas classes utilizando *kernel* RBF. 79
6.3 Ranking dos algoritmos de classificação: Twin-KSCV, LSTKSVC e o algoritmo proposto, utilizando *kernel* RBF. 81
6.4 Comparativo entre as acurácias obtidas entre vários algoritmos encontrados na literatura. 82

7.1 Subconjuntos baseado no ranking do *F-score*. 86
7.2 Subconjuntos baseado no ranking do $\chi^2$. 86
7.3 Medidas de desempenho detalhadas com os melhores resultados, modelo calculado - SVM. 87
7.4 Classificação de importância das variáveis segundo $\chi^2$ e RFI. 90
7.5 Melhores resultados (em %) obtidos com SVM, MLP e ELM associado com seleção de variáveis. 91
7.6 Medidas de desempenho detalhadas com os melhores resultados (em %). 95
7.7 Descrição das bases de dados utilizadas. 97
7.8 Comparação de desempenho de conjuntos de dados com algoritmos usando o *kernel* RBF. 100
7.9 Valores obtidos da acurácia, sensitividade e especificidade utilizando o FSTBSVM com *kernel* RBF. 101
7.10 Comparativo entre as acurácias obtidas, utilizando o conjunto de dados *BCWO*. 102
7.11 Comparativo entre as acurácias (em %) obtidas pelo SVM, MLP e FSTBSVM nas amostras de laranjas. 104
7.12 Comparativo entre as acurácias (em %) obtidas pelo SVM, MLP, RF e FSTBSVM nas amostras de tomate. 104
7.13 Comparativo entre as acurácias (em %) obtidas pelo SVM, MLP, RF e FSTBSVM nas amostras de pimentão. 105
7.14 Comparativo entre as acurácias (em %) obtidas pelo SVM, MLP, RF e FSTBSVM nas amostras de cebola. 106
7.15 Comparativo entre as acurácias (em %) obtidas pelo SVM, MLP, RF e FSTBSVM nas amostras de alface. 107
7.16 Comparativo entre as acurácias (em %) obtidas pelo SVM e FSTBSVM nas amostras de vinhos produzidos das uvas syrah. 108

# Lista de Siglas

| | |
|---|---|
| *SVM* | *Support Vector Machine* |
| *PSVM* | *Proximal Support Vector Machine* |
| *GEPSVM* | *Generalized Proximal Support Vector Machine* |
| *TWSVM* | *Twin Support Vector Machine* |
| *TBSVM* | *Twin Bounded Support Vector Machine* |
| *QPP* | *Quadratic Programming Problem* |
| *RFE* | *Recursive Feature elimination* |
| *LSPTSVM* | *Least Squares Recursive Projection Twin Support Vector Machine* |
| *FLSPTSVM* | *Feature Selection Least Squares Projection Twin Support Vector Machine* |
| $K-SVCR$ | *Support Vector Classification-Regression for K-class* |
| *LSTSVM* | *Least Squares Twin Support Vector Machine* |
| $FS-LSTSVR$ | *Feature Selection Least Squares Twin Support Vector Regression* |
| *LSTSVR* | *Least Squares Twin Support Vector Regression* |
| $LST-KSVC$ | *Least Squares Twin KSVC* |
| $ICP-MS$ | *Inductively Coupled Plasma Mass Spectrometry* |
| *SOR* | *Successive Over-Relaxion* |
| *RBF* | *Radial Basis Function* |
| *KKT* | *Karush-Kuhn-Tucker* |
| *ITSVM* | *Improved Twin Support Vector Machine* |
| *NHSVM* | *Nonparallel Hyperplane Support Vector Machine* |
| *LPTWSVM* | *Linear Programming Twin Support Vector Machine* |
| *MLPTWSVM* | *Multi-Class Linear Programming Twin Support Vector Machine* |
| *NPPC* | *Nonparallel Plane Proximal Classifier* |
| *GTWSVM* | *Twin Support Vector Maquine via Fast Generalized Newton Refinement* |
| *KBLSTWSVM* | *Knowledge Based Least Squares Twin Support Vector Maquine* |
| $ODAG-LSTSVM$ | *multi-class LSTSVM Classifier Based on Optimal Directed Acyclic Graph* |
| *SVR* | *Support Vector Regression* |
| *TSVR* | *Twin Support Vector Machine for Regression* |
| *TLSSVR* | *Twin Least Squares Support Vector Machine for Regression* |
| *WLSTSVM* | *Weighted Least Squares Twin Support Vector Machine* |
| $UNH-MTSVM$ | *Margin-based Twin Support Vector Machine with Unity Norm Hyperplanes* |
| *MBSVM* | *Multiple Birth Support Vector Machine* |
| *DTTSVM* | *Decision Tree Twin Support Vector Machine* |
| *NELSTSVM* | *1-norm Least Squares Twin Support Vector Machine* |
| $Lap-LSTSVM$ | *Laplaciano Least Squares Twin Support Vector Machine* |
| *MLTSVM* | *Multi-Label Twin Support Vector Machine* |
| $\nu-SVM$ | $\nu$- *Support Vector Machine* |
| *WLTSVM* | *Weighted Twin SVM with Local Information and its application* |
| $S-TWSVM$ | *Structural TWSVM* |

| | |
|---|---|
| *MVSVM* | *Multi-Weight Vector Projection Support Vector Machine* |
| *PTSVM* | *Projection Twin Support Vector Machine* |
| *RPTSVM* | *Regularization Projection Twin Support Vector Machine* |
| *SRPTSVM* | *Structural Regularization Projection Twin Support Vector Machine* |
| *EMVSVM* | *Enhanced MVSVM* |
| *TPMSVM* | *Twin parametric-margin SVM* |
| *STPMSVM* | *Smooth Twin Parametric-Margin Support Vector Machine* |

# Lista de Notações

| | |
|---|---|
| $\mathbf{x}^T$ | *Transposto do vetor* $\boldsymbol{x}$ |
| $\mathbb{R}^n$ | *Espaço n-dimensional* |
| $\sum_{i=1}^{n}$ | *Soma de elementos de 1 até n* |
| $\partial$ | *Derivada parcial* |
| $\nabla_{\mathbf{x}} L$ | *Gradiente de L em relação a* $\boldsymbol{x}$ |
| $\| \mathbf{x} \|$ | *Norma do vetor* $\boldsymbol{x}$ |
| $\mathbf{w}^T \mathbf{x}$ | *Produto escalar entre o vetor* $\boldsymbol{w}$ *e o vetor* $\boldsymbol{x}$ |
| *s.a.* | *sujeito à* |
| min | *mínimo* |
| max | *máximo* |
| *sgn* | *sinal* |
| $\xi$ | *variável de folga* |
| $K(\mathbf{x}, \mathbf{y})$ | *função kernel* |
| $\int$ | *integral* |
| $\lim_{n \to \infty}$ | *limite com n tendendo ao infinito* |
| *tanh* | *tangente hiperbólica* |
| *e* | *Vetor composto por elementos* "1" *com dimensão apropriada* |

CAPÍTULO 1

# Apresentação

Aprendizado de máquina, do inglês *machine learning*, surge do estudo de reconhecimento de padrões e da teoria da aprendizagem computacional em inteligência artificial [105]. Em 1959, Arthur Samuel desenvolveu, provavelmente, a primeira pesquisa de aprendizagem de máquina bem sucedida, na qual define o aprendizado de máquina como o campo de estudo que dá aos computadores a habilidade de aprender, sem serem explicitamente programados [107, 134].

Aprendizagem de máquina é o campo de estudo científico em que se concentram os algoritmos de indução e outros algoritmos que podem aprender [126]. Tais algoritmos operam construindo um modelo, a partir dos dados de entrada, de modo a gerar classificadores para um conjunto de exemplos [95]. A classificação é uma das mais importantes e populares tarefas da mineração de dados, esse processo de classificação consiste em atribuir um rótulo de classe a uma determinada informação [134].

Com uma considerável ascensão na última década, o algoritmo intitulado Máquina de Vetores de Suporte, que vem do inglês, *Support Vector Machine* (SVM), introduzido em 1992 por Boser et al. [13], e em 1995 por Vapnik e Cortes [163], tem se apresentado superior quando comparado a outros algoritmos de aprendizado de máquina [105, 171]. SVM é um classificador baseado em funções *kernel*. Proposto inicialmente como um classificador binário que otimiza um hiperplano entre duas classes de amostras, a fim de maximizar a margem de separação entre elas. Sempre que as classes não são linearmente separáveis, diferentes funções *kernel* podem transformar o espaço de características em um espaço de alta dimensão, onde as classes podem ser separadas de forma linear [30]. O SVM tem se apresentado bem eficaz em várias áreas, tais como: reconhecimento de padrões, reconhecimento de objetos, identificação de fala, detecção de imagens de face e categorização de texto [15].

## 1.1 Motivação

Com o passar dos anos, o SVM e algumas de suas extensões estão sendo aprimoradas por pesquisadores de todo o mundo, com aplicações nas mais variadas

áreas: na saúde, com o diagnóstico de câncer de mama [2]; na área de alimentos com a classificação de vinhos [31, 68], reconhecimento de bebida e vinagre [91], origem geográfica de arroz [96], na classificação de alimentos cultivados nos sistemas orgânicos e convencional [96], na área de economia, finanças e tendência no mercado de ações [83], entre outros.

Motivados pelos estudos realizados com o algoritmo SVM, realizamos a classificação de alimentos cultivados nos sistemas orgânicos e convencionais, e também a classificação de vinhos produzidos em países distintos, visto que os algoritmos SVMs têm se mostrado bem eficazes como técnica de classificação, podendo ser usados como ferramenta na detecção de fraudes por órgãos ou agências de regulação desses tipos de alimentos.

Nos últimos anos, a estrutura básica do SVM foi alterada por novas extensões para aumentar a robustez e precisão de classificação. Fung e Mangasarian [43], realizando uma modificação em uma das desigualdades das restrições do problema do SVM, propuseram o algoritmo denominado *Proximal Support Vector Machine* (PSVM). Mangasarian e Wild [101] não exigiram que os hiperplanos fossem paralelos, mas que cada hiperplano estivesse o mais próximo possível a um agrupamento de dados e o mais longe possível do outro agrupamento de dados. Esse algoritmo foi denominado *Generalized Eigenvalues Proximal Support Vector Machine* (GEPSVM). Jayadeva et al. [66], motivados pelo GEPSVM [101] e utilizando a formulação do SVM, apresentaram o *Twin Support Vector Machine* (TWSVM). A ideia do TWSVM é construir dois planos não paralelos, no qual cada plano é determinado pela resolução de um problema de programação quadrática (QPP), ou seja, resolver dois problemas ao invés de um, conforme é feito no SVM. A eficiência do TWSVM como classificador foi relatada em algumas pesquisas, como no diagnóstico de doenças do coração [154], diagnóstico de Alzheimer [3], detecção de cluster de microcalcificação em mamografias [189, 190, 52], detecção de doenças e análise de imagens [188], diagnóstico de diabetes em mulheres grávidas [155], classificação de gesto, baseado em Eletromiograma (EMG) [108, 5], reconhecimento de fala [27, 92, 181], reconhecimento de padrões [66, 109, 120], reconhecimento de emoção [159] e detecção de intruso [57, 40]. Várias extensões e variações do TWSVM têm sido proposta até os dias atuais, dentre elas o *twin bounded support vector machine* (TBSVM) proposto por Shao et al. [146].

Devido ao crescente aumento de pesquisas relacionadas ao TWSVM e suas extensões, pudemos notar um número pequeno de estudos em que os algoritmos trabalham na resolução de problemas de múltiplas classes, bem como existem poucos estudos utilizando as ideias do TWSVM como seletor de variáveis. Assim, propomos um algoritmo para problemas de classificação de múltiplas classes, utilizando a estratégia "$1-\times-1-\times-$ *restante*" com uma versão mínimos quadrados, em que será necessá-

rio resolver sistemas de equações lineares, ao invés de dois problemas de programação quadrática. Nesse algoritmo foi implementando o princípio de minimização do risco estrutural que auxilia na capacidade de generalização. Com esse mesmo viés, apresentamos um algoritmo seletor de variáveis baseado no TBSVM.

## 1.2 Objetivos

### 1.2.1 Objetivo Geral

O objetivo desta tese é propor um algoritmo de classificação para problemas de múltiplas classes envolvendo uma extensão do TWSVM.

### 1.2.2 Objetivos Específicos

Para alcançar este objetivo geral, alguns objetivos específicos foram estabelecidos:

1. Descrever o estado da arte dos algoritmos relacionados às máquinas de vetores de suportes (SVMs), máquinas de vetores de suportes duplos (TWSVMs) e suas extensões.
2. Utilizar algoritmos de classificação em bancos de dados relacionados a alimentos cultivados nos sistemas orgânicos e convencionais e classificação quanto à origem geográfica de vinhos, cujas uvas foram cultivadas em países distintos.
3. Desenvolver um seletor de variáveis baseado no TBSVM.

## 1.3 Organização da Tese

A organização da tese se encontra do seguinte modo: no Capítulo 2 descrevemos o referencial teórico em relação ao SVM, TWSVM e suas extensões. No Capítulo 3 fazemos um detalhamento sobre teoria matemática por trás das SVMs, para o caso linear e não linear. Apresentamos o Teorema de Mercer e função *kernel*. Algumas extensões do SVM, tais como PSVM e GEPSVM serão apresentadas no final do capítulo. No Capítulo 4 introduzimos o TWSVM para o caso linear e não linear. Uma extensão do TWSVM, denominada TBSVM também é apresentada nesse capítulo. No Capítulo 5 apresentamos alguns trabalhos sobre máquinas de vetores de suportes para múltiplas classes, que foram a motivação para a proposta do próximo capítulo. No Capítulo 6 desenvolvemos um algoritmo com melhorias na versão mínimos quadrados para problemas de múltiplas classes, baseado no TWSVM (Algoritmos 6.2.1 e 6.2.2). No Capítulo 7 serão apresentadas aplicações em bancos de dados reais, relativos às amostras de laranjas e vegetais (alface,

tomate, cebola e pimentão), em relação a classificação quanto a forma de cultivo, amostras de vinhos para classificação quanto a origem geográfica e um seletor de variáveis baseado em uma extensão do TWSVM. Por fim, no Capítulo 8 são apresentadas as conclusões gerais, e apontamos alguns trabalhos futuros.

CAPÍTULO 2

# Referencial Teórico

## 2.1 Considerações Iniciais

Neste capítulo, são apresentadas os principais estudos sobre as máquinas de vetores de suportes (SVMs), máquinas de vetores de suportes duplos (TWSMV) e suas extensões.

## 2.2 SVM

Fundamentada na teoria de aprendizado estatístico desenvolvida por Vladimir Vapnik [163], na qual foi estruturada a teoria matemática para a escolha de um ótimo classificador, e usando o princípio indutivo de minimização do risco estrutural, em relação ao conjunto de amostras de treinamento e teste [95], o SVM é uma poderosa ferramenta para aprendizado supervisionado e amplamente usado em problemas de classificação e regressão [79]. O SVM pode ser entendido como uma superfície que define um limite entre várias amostras de dados em um espaço multidimensional de acordo com o valor de suas variáveis [81].

Para o caso linear, o SVM busca determinar um hiperplano de margem máxima que separa ao máximo as amostras de treinamento [164]. Há situações em que os dados de treinamento não são linearmente separáveis, neste caso são inseridas variáveis de folga na função objetivo do problema de otimização do SVM, que funciona como uma penalização para as amostras que foram classificadas erroneamente, do lado contrário do hiperplano [30]. Essas situações são para exemplos simples, ou para fins didáticos. Na maioria dos dados reais não é possível determinar esse hiperplano sem antes utilizar o truque do *kernel* [137]; o *kernel* é uma função que mapeia as amostras de treino em um espaço de alta dimensão, de tal modo que as amostras possam ser separadas linearmente por um hiperplano nesse novo espaço. As funções *kernel* são usadas como um truque (truque do *kernel*), permitindo a separação linear por um hiperplano entre amostras que não podem ser separadas por um hiperplano no espaço de atributos original. Devido ao seu grande

desempenho de generalização, o SVM tem sido aplicado com êxito em vários aspectos, desde aprendizado de máquinas, mineração de dados e descoberta de conhecimento.

Em 1998, Burges [15] relatou algumas áreas de aplicação em que o SVM se apresenta bem eficiente, tais como: reconhecimento de padrões, reconhecimento de objetos, identificação de fala, detecção de imagens de face e categorização de texto. Com o passar dos anos, o SVM e algumas de suas extensões estão sendo aprimoradas por pesquisadores de todo o mundo, com aplicações nas mais variadas áreas: na saúde, com o diagnóstico de câncer de mama [2]; na área de alimentos com a classificação de vinhos [32, 68], reconhecimento de bebida e vinagre [91], origem geográfica de arroz [96], na classificação de alimentos cultivados nos sistemas orgânicos e convencional [162], na área de economia, finanças e tendência no mercado de ações [83], entre outros.

## 2.3 PSVM

Fung e Mangasarian [43] propuseram, em 2001, uma modificação em uma das desigualdades das restrições do problema do SVM, trocando por uma igualdade, esse novo algoritmo foi denominado *Proximal Support Vector Machine* (PSVM). O PSVM diferencia-se do SVM em sua formulação e comportamento, pois agora o interesse é classificar amostras que estejam mais próximas de um dos hiperplanos paralelos e o mais distante possível do outro hiperplano.

## 2.4 GEPSVM

Posteriormente, em 2006, Mangasarian e Wild [101] propuseram um novo algoritmo que não exigia que os hiperplanos fossem paralelos, mas que cada hiperplano estivesse o mais próximo possível a um agrupamento de dados e o mais longe possível do outro agrupamento de dados. Esse novo método foi denominado de *Generalized Eigenvalues Proximal Support Vector Machine* (GEPSVM).

## 2.5 TWSVM

Jayadeva et al. [66], motivados pelo GEPSVM [101] e utilizando a formulação do SVM, propuseram em 2007 o *Twin Support Vector Machine* (TWSVM). A ideia do TWSVM é construir dois planos não paralelos, em que cada plano é determinado pela resolução de um problema de programação quadrática (QPP), ou seja, resolver dois problemas ao invés de um, conforme é feito no SVM. Uma das vantagens do TWSVM é em relação ao tempo, pois, para resolver um problema com uma quantidade $l$ de amostras,

o tempo gasto utilizando o TWSVM é quatro vezes menor, quando comparado com o tradicional SVM [66].

A eficiência do TWSVM como classificador foi relatada em algumas pesquisas, como no diagnóstico de doenças do coração [154], detecção de cluster de microcalcificação em mamografias [189, 190, 52], detecção de doenças e análise de imagens [188], diagnóstico de diabetes em mulheres grávidas [155], classificação de gesto, baseado em Eletromiograma (EMG) [108, 5], reconhecimento de fala [27, 92, 181], reconhecimento de padrões [66, 109, 120], reconhecimento de emoção [159] e detecção de intruso [57, 40].

Várias extensões e variações do TWSVM têm sido proposta até os dias atuais, Kumar e Gopal [78] propuseram uma melhoria no algoritmo TWSVM, e resolve o problema de programação quadrática primal convertendo em um problema de minimização sem restrições, seguindo as ideias propostas por Lee e Mangasarian [84].

Shao et al. [146] implementaram o princípio de minimização do risco estrutural ao TWSVM, ao adicionar um termo de regularização [164], combinado com o método *successive over-relaxion* (SOR) [99] para resolução do problema de programação quadrática e assim melhorar o tempo computacional e essa extensão foi denominada *twin bounded support vector machine* (TBSVM). Ao introduzir diferentes funções lagrangianas para o problema primal do TBSVM, Yingjie et al. propuseram o *improved twin support vector machine* (ITSVM) [152] o qual supera a desvantagem de ter que resolver inversão de matrizes que, para bancos de dados com grandes número de amostras, se torna impraticável.

## 2.6 Mínimos quadrados

Kumar e Gopal propuseram, em 2009, *least squares twin support vector machine* (LSTSVM) [79], nesse trabalho a versão mínimos quadrados é proposta para o TWSVM, e resolve dois problemas primais ao invés de dois problemas duais. A resolução dos problemas primais se resume na resolução de dois sistemas de equações lineares, esse algoritmo melhora a eficiência da precisão e tempo computacional, mas apresenta a desvantagem de ser sensível a ruídos e sofre com o problema de super ajustamento [157]. Chen e Ji [21] prosuseram uma melhoria no LSTSVM denominada *weighted least squares twin support vector machines* (WLSTWSVM), em que diferentes pesos são atribuídos às variáveis a fim de eliminar o impacto desses ruídos e obter uma classificação mais robusta.

Utilizando as ideias do LSTSVM, combinadas com a versão *least squares* para o SVM (LSSVM), Shao et al. [144] apresentaram o *least squares projection twin support vector machine* (LSPTSVM), em que um termo de regularização extra é inserido, garantindo melhor habilidade de generalização. Tomar et al. [160] estenderam a versão WLSTWSVM para problemas de classificação binária em conjunto de dados não balan-

ceados. Uma extensão do LSPTSVM, combinada com WLSTWSVM em problemas de classificação binária, é proposta por Hua e Ding [62] e mede a importância dos pontos que compartilham a mesma classe de rótulos. Assim, uma ponderação média é atribuída ao invés de uma média padrão comumente usada, refletindo numa estrutura geométrica melhor.

Kumar et al. [80] propuseram *knowledge based least squares twin support vector machine* (KBLSTWSVM), esse algoritmo supera em eficiência e rapidez a versão proposta para o SVM [42] e PSVM [69]. Uma mudança da norma-2 pela norma-1 é feita por Gao et al. [44], que introduz o termo de regularização Tikhonov [153], frequentemente usado para regularizar mínimos quadrados. Dessa forma um novo método que realiza seleção de variáveis e classificação é proposto, denominado *1-norm least squares twin support vector machine* (NELSTSVM). Com objetivo de adaptar o método NELSTSVM, Zhang et al. [191] propuseram $l_p$*-norm* LSTSVM $(0 < p < 1)$, onde o parâmetro $p$ pode ser automaticamente escolhido para cada conjunto de dados, utilizando assim a norma-$p$.

O princípio de minimização do risco estrutural é implementado por Xu et al. [179] na versão LSTSVM, com o intuito de melhorar a habilidade de classificação. Em Shao et al. [144], um termo de regularização extra é adicionado à função objetivo do problema primal de [25], a fim de melhorar a capacidade de generalização, mas isso é feito somente para o caso linear. A ampliação para o caso não linear é feita por Ding e Hua [38]. A ampliação da versão mínimos quadrados para casos de regressão são propostas por Zhao et al. [192], Huang et al. [63] e Huang et al. [64]. A ideia do TWSVM foi estendida para aprendizado semi-supervisionado por Chen et al. [23] denominado por *laplaciano least squares twin support vector machine* (Lap-LSTSVM). Um seletor de variáveis baseado no LSTSVM foi proposto por Tomar e Agarwal [154], já Guo et al. [51] propuseram um seletor de variáveis baseado no LSPTSVM. Na área de visão computacional, o LSTSVM é aplicado ao reconhecimento de atividades humana em Mozafari et al. [106], Nasiri et al. [110] e Khemchandani e Sharma [74].

Ghorai et al. [46] propuseram o *nonparallel plane proximal classifier* (NPPC) para problemas de classificação binária, que pode ser interpretado como um classificador obtido por otimizar o erro quadrático médio regularizado, assim resolve dois sistemas de equações lineares ao invés de resolver dois problemas de programação quadrática, conforme é previsto na versão original do TWSVM. Uma reformulação do NPPC é feita em Ghorai et al. [45] o qual inclui as restrições $\| w_i \| = 1$ com $i = 1,2$, que não são consideradas em Ghorai et al. [46] e Jayadeva et al. [66]. Porém as inclusões dessas restrições fazem o problema se tornar não linear, e o reduz a um problema de otimização convexa do tipo mínimos quadrados. Para resolver esse problema, é utilizado o método de convergência quadrática de Newton, com o tamanho do passo definido pela busca de Armijo, fazendo que o problema seja resolvido de modo iterativo e com um número

finito de passos. A estratégia de adicionar as restrições $\| w_i \| = 1$ com $i = 1,2$ também é feita em *a novel margin-based twin support vector machine with unity norm hyperplane* (UNH-MTSVM) [143], e a minimização do risco estrutural é implementada por adicionar o termo de regularização, conforme foi feito em Shao et al. [146].

Em contraste aos modelos não paralelos citados acima, o *nonparallel hyperplane support vector machine* (NHSVM), proposto por Shao et al. [139], constrói simultaneamente os dois hiperplanos resolvendo um único problema de programação quadrática, utilizando a técnica SOR. Qi et al. [129] propuseram uma extensão à versão $\mathfrak{U}$-SVM [170], em que dois hiperplanos não paralelos são determinados, denominado por $\mathfrak{U}$-NSVM para problemas de aprendizagem supervisionada com dados universais, e apresentaram a vantagem de ter poucos parâmetros e boa capacidade de generalização.

## 2.7 ν-TWSVM

Similar ao realizado por Schölkopf et al. [136], que propôs o ν-*support vector machine* (ν-SVM), em que o parâmetro ν é inserido e determina um limite superior sobre os erros de treinamento e um limite inferior sobre o número de vetores de suporte, uma extensão do ν-SVM para o TWSVM é proposta por Peng [116], denominada por ν-*twin support vector machine* (ν-TSVM), melhorando a taxa de precisão no processo de classificação. Porém o (ν-TSVM) não leva em consideração a importância de cada amostra enquanto constrói o hiperplano de separação durante a fase de aprendizagem. Xu e Guo [176] estende a versão ν-TSVM implementando o princípio de minimização do risco estrutural, a fim de minimizar os erros durante a fase de treinamento, garantindo uma melhor precisão. Khemchandani et al. [73], motivados pelos trabalhos citados, propuseram dois algoritmos (*Iν-TWSVM* e *Iν-TWSVM (Fast))*, enquanto o primeiro melhora a habilidade no processo de classificação, o segundo se apresenta mais eficiente em tempo computacional.

Utilizando as ideias do ν-SVM, um novo algoritmo foi proposto em Hao [55], que visa encontrar um modelo de margem paramétrica de forma arbitrária, muito útil para dados heteroscedásticos. Uma extensão desse trabalho é apresentada por Peng [118], utilizando as ideias do TWSVM, intitulada *twin parametric-margin SVM* (TPMSVM). O objetivo foi gerar dois hiperplanos não paralelos, de forma que cada um determinasse o hiperplano de margem paramétrica positiva ou negativa do hiperplano de separação. Uma melhoria no TPMSVM foi proposta por Wang et al. [169], denominado *smooth twin parametric-margin support vector machine* (STPMSVM), que propõe uma versão primal, com o intuito de resolver dois problemas de programação quadrática no espaço primal por convertê-los em dois problemas de minimização sem restrições, ao invés de dois problemas de programação quadrática no espaço dual, conforme foi feito em Peng [118]. O

método de Newton, com tamanho de passo definido pela busca de Armijo, foi empregado para resolução do problema e foi utilizado também um algoritmo genético a fim de selecionar ótimos parâmetros, bem como fornecer seleção de variáveis discriminante. Peng et al. [119] propuseram uma melhoria no TPMSVM, e geram funções de decisões esparsas e também otimiza simultaneamente as projeções dos centróides das duas classes em cada problema de otimização. Uma extensão do TPMSVM para problemas com dados semi-supervisionado foi apresentada por Yang e Xu [184], que utiliza a regularização laplaciana. Com o intuito de melhorar a sensitividade em relação aos *outliers* no TWSVM, Xie e Sun [173] propuseram o *multitask centroid twin support vector machine*.

Interessados em explorar as informações entre amostras de mesma classe e classes distintas e por acreditarem que amostras de mesma classe têm maior similaridade, Ye et al. [185] sugeriram *weighted twin support vector machines with local information and its application* (WLTSVM), que reformula o problema primal do TWSVM e utiliza a teoria do grafo espectral na construção de um grafo com $k$-vizinhos mais próximos para modelar a estrutura geométrica local. Xu et al. [178] propuseram uma melhoria na versão do ν-TSVM, denominada *rough* ν*-TSVM*, no qual diferentes penalidades são dadas às amostras negativas, e as amostras positivas compartilharam das mesmas ponderações, ou seja, as informações locais entre amostras positivas não são exploradas. A fim de garantir que as amostras positivas recebam penalidades que contribuam para a construção do hiperplano dessas amostras, Xu et al. [180] sugeriram *KNN- based weighted rough* ν*-twin support vector machine*. Qi et al. [128] apresentaram o *structural TWSVM* (*S*-TWSVM), que extrai as informações estruturais e utilizam no método *cluster* hierárquico, construindo um classificador mais robusto. Utilizando a proposta feita por Xu et al. [180], Pan et al. [111] aplicam o truque do $k$-vizinho mais próximo no *S*-TWSVM, onde diferentes pesos são aplicados às amostras entre os *clusters* e removem as restrições redundantes, garantindo assim, que o processo ganhe em tempo computacional e precisão.

Para problemas não balanceados, ou seja, problemas em que o número de amostras de uma classe é superior ao da outra classe, o TWSVM apresenta a mesma fraqueza que o SVM, e tem sido intensivamente estudado [149, 166, 194]. A fim de melhorar o poder de classificação para dados não balanceados, Shao et al. [142] realizaram uma extensão do TWSVM denominada *weighted lagrangian twin support vector machine* (WLTSVM), onde uma ponderação é introduzida para controlar a sensitividade em relação à proporção desbalanceada na construção de cada hiperplano. Uma extensão do WLTSVM é proposta também por Shao et al. [141] em que uma função linear ponderada de perda é introduzida e os dois sistemas de equações lineares são resolvidos, utilizando o algoritmo gradiente conjugado, melhorando ainda mais a sua capacidade de generalização.

Para explorar a informação local em cada classe, em problemas de classificação

e regressão, Xu et al. [175] propuseram *KNN-based weighted multi-class twin support vector machine*. Eles atribuíram pesos diferentes para amostras de mesma classe e introduzindo funções $f$ e $h$ nas restrições para explorar informações intra-classe e extra-classe.

## 2.8 Projeção

Atribuindo um vetor de pesos, obtidos ao resolver um par de problemas de autovalores, tais que as amostras de uma classe estejam o mais próximo possível de sua classe, e o mais distante da outra classe, Ye et al. [186] propuseram o *multi-weight vector projection support vector machine* (MVSVM). Inspirado no MVSVM, Chen et al. [25] sugeriram *projection twin support vector machine* (PTSVM), que resolve dois problemas, obtendo duas direções de projeção, semelhante ao TWSVM. Shao et al. [145] introduziu um termo de regularização ao PTSVM, superando, assim, as dificuldades apresentadas pelos problemas de singularidade de matriz no PTSVM, esse algoritmo foi denominado *regularization PTSVM* (RPTSVM).

Motivados pelo PTSVM e RPTSVM, Peng e Xu [121] apresentaram o *structural RPTSVM* (SRPTSVM), construindo um conjunto de restrições para cada problema de otimização, melhorando a capacidade de generalização por implementar um par de termos de regularização de margem máxima, e o princípio de minimização do risco estrutural. Recentemente, Gu et al. [49] fizeram a mudança da norma-2 para norma-1 no *enhanced MVSVM* (EMVSVM), que é uma extensão do MVSVM, obtendo um algoritmo mais robusto contra ruídos e *outliers*.

Uma versão denominada *twin support vector machine via fast generalized Newton refinament* (GTWSVM) foi proposta por Wang et al. [167], o qual supera em tempo computacional quando comparado com o TWSVM e o SVM.

## 2.9 Múltiplas classes

As extensões/variações citadas até agora, do TWSVM, são adequadas para problemas de classificação binária, porém problemas de classificação de múltiplas classes são muito frequentes no dia a dia. Chen e Ji [20] apresentaram uma versão para problemas de múltiplas classes, motivados pelo trabalho apresentado por Platt et al. [124]. O algoritmo denominado *multi-class LSTSVM classifier based on optimal directed acyclic graph* (ODAG-LSTSVM) foi proposto e apresentou alto desempenho de classificação e tempo computacional em comparação com a versão proposta para o SVM [124]. Motivado por Angulo et al. [4], que apresentaram um algoritmo de classificação para múltiplas classes denominado K-SVCR. Xu et al. [177] propuseram *a twin multi-class classification sup-*

*port vector machine* (Twin-KSVCR), algoritmo que combina a velocidade do TWSVM, com as vantagens do K-SVCR, pois esse algoritmo de aprendizagem com saídas ternárias $\{-1,0,1\}$ é baseado na teoria do SVM e avalia todas as amostras em uma estrutura $\{1-\times-1-\times-restante\}$ na fase de decomposição, usando uma combinação de classificação e regressão.

Nasiri et al. [109] estende o estudo apresentado por Xu et al. [177], combinado com o K-SVCR [4], para a versão mínimos quadrados denominada *least squares twin multi-class classification support vector machine* (LST-KSVC), apresentando uma vantagem em tempo computacional, pois resolve dois sistemas de equações lineares, ao invés de dois problemas de programação quadrática, conforme feito no Twin-KSVCR [177] e elevado poder de classificação em problemas de múltiplas classes por utilizar uma estrutura $\{1-\times-1-\times-restante\}$. Chen e Xu [22] sugeriram uma outra extensão para problemas de múltiplas classes utilizando a versão mínimos quadrados e acrescentaram termos de regularização a fim de controlar a complexidade, pois encontram hiperplanos adequados, garantindo que cada hiperplano está mais próximo de sua própria classe e o mais longe possível da outra classe. Eles utilizam uma estrutura $\{1-\times-todos\}$, de forma que esse algoritmo resolve $k$ problemas de programação quadrática de tamanhos menores, considerando $k$ o número de classes.

Tomar e Agarwal [156] apresentaram 4 algoritmos combinando a rapidez e eficiência de classificação da versão mínimos quadrados, utilizando as estruturas $\{1-\times-todos\}$, $\{1-\times-1\}$, $\{todos-\times-1\}$ e grafo acíclico dirigido. São também apresentadas as vantagens e desvantagens de cada uma dessas estruturas e confirma que a estrutura grafo acíclico dirigido, dentre as bases de dados testados, foi o classificador com melhor desempenho. Em outro estudo, Tomar e Agarwal [158] estenderam a versão do TWSVM para múltiplas classes utilizando a estrutura $\{1-\times-1\}$, e a técnica SOR para resolução dos problemas de programação quadrática e assim garantir velocidade na fase de treinamento. Utilizando uma estrutura $\{1-\times-todos\}$, Chen et al. [24] apresentam o *multi-label twin support vector machine* (MLTSVM) para problemas de classificação de múltiplas classes, e resolve os problemas de programação quadrática utilizando a técnica SOR.

Li e Tian [85] propuseram dois algoritmos, sendo um para classificação binária, denominado *linear programming twin support vector machine* (LPTWSVM), e outro para problemas de múltiplas classes denominado *multi-class linear programming twin support vector machine* (MLPTWSVM), que realiza a mudança da norma-2 para norma-1 na função objetivo e introduz o termo de regularização a fim de implementar o princípio de minimização do risco estrutural. Outra vantagem apresentada por esse algoritmo é a conversão do problema primal para programação linear, evitando o cálculo de matrizes inversas, que computacionalmente é muito lento.

Yan et al. [183] apresentaram uma extensão do TWSVM [66] para problemas de múltiplas classes denominado *multiple birth support vector machine* (MBSVM), que resolve simultaneamente múltiplos problemas de programação quadrática e se destaca por apresentar baixa complexidade computacional, porém, ao construir os classificadores, alguns pontos podem causar ambiguidade na classificação. Para superar esses inconvenientes Shao et al. [140] propuseram *the best separating decision tree twin support vector machine* (DTTSVM), que constrói árvores binárias para classificação em problemas de múltiplas classes.

## 2.10 Regressão

Para problemas de regressão, *twin support vector maachine for regression* (TSVR) foi proposto por Xingun [117], inspirado no *support vector regression* (SVR) [147] e TWSVM, apresenta boa capacidade de generalização e baixo tempo computacional. Khenchadani et al. [71] também motivados pelo SVR e [10], propuseram o TWSVR e comentam que a essência desse método proposto segue o mesmo padrão do TWSVM, diferentemente do TSVR. Peng et al. [122] apresentaram um novo algoritmo para regressão, que introduz um par de novos termos no problema de otimização, a fim de encontrar dois eixos de projeção para os pontos de treinamento, incorporando a informação estrutural dos dados no processo de treinamento.

Combinando a velocidade computacional do TWSVM, com a capacidade de generalização do LSTWSVM em problemas de regressão, Zhao et al. [192] apresentaram o *twin least squares support vector regression*. Huang et al. [63] combinaram o TSVR com a versão mínimos quadrados, resolvendo LSTSVR diretamente no espaço primal, ao invés do espaço dual. Combinando a estrutura dos dados de treinamento com suas distribuições, Parastalloi et al. [112] sugeriram o *modified twin support vector regression* (MTSVR), e utilizam o algoritmo SOR para resolver os problemas de programação quadrática, melhorando assim o tempo computacional, e utiliza o *particle swarm optimization* (PSO) para determinar os parâmetros ótimos para o modelo MTSVM.

## 2.11 Seletor de variáveis

Um seletor de variáveis baseado no LSTSVR (FS-LSTSVR) é apresentado por Huang et al. [64], onde adicionam o termo de regularização de Tikhonov na função objetivo do LSTSVR, garantindo que os dois problemas de programação quadrática sejam fortemente convexo, nesse estudo a formulação do FS-LSTSVR é convertida em problemas de programação linear, fazendo a mudança da norma-2 para norma-1 na função objetivo do LSTSVR.

## 2.12 Aplicações

Embora a mamografia assistida por computador venha sendo muito estudada nas duas últimas décadas, a detecção automatizada de sinais de micro-calcificação ainda permanece muito difícil, nesse sentido Zhang [188] propôs uma abordagem baseada no *boosted twin support vector machine* para detecção de agrupamento de micro-calcificação em imagens de mamografias digitais, obtendo uma sensitividade (>92%). Zhang e Gao [189], em um outro trabalho intitulado *twin support vector machine and subspace learning methods for microcalcification cluster detection*, incorporam os algoritmos de aprendizagem subespaço combinado com TWSVM e melhoraram a sensitividade para mais de (>94%).

A fim de classificar e identificar a contração do músculo, durante a flexão de dedos usando um conjunto de variáveis fractal, que envolvem dimensão fractal e comprimento máximo fractal, por sinal de eletromiograma acoplados por dois canais nos músculos do antebraço, Arjunan et al. [5] propuseram a aplicação dessa metodologia combinada com o TWSVM, obtendo satisfatórios resultados, sugerindo ser uma combinação adequada para aplicação em próteses.

A eficiência do TWSVM como classificador foi apresentada também no diagnóstico de doenças do coração [154], diagnóstico de Alzheimer [3], detecção de cluster de microcalcificação em mamografias [189, 190, 52], detecção de doenças e análise de imagens [188], diagnóstico de diabetes em mulheres grávidas [155], classificação de gesto, baseado em Eletromiograma (EMG) [108, 5], reconhecimento de fala [27, 92, 181], reconhecimento de padrões [66, 109, 120], reconhecimento de emoção [159] e detecção de intruso [57, 40], classificação de sinais de eletroencefalograma (EEG) para o diagnóstico de distúrbios neurológicos e detecção de crises [133], dentre outros.

CAPÍTULO 3

# Máquinas de Vetores de Suporte (SVMs) e extensões

## 3.1 Considerações Iniciais

Neste capítulo, apresentamos a construção teórica do SVM para os casos linearmente separáveis com margens rígidas e suaves, o caso não linearmente separável, a definição da função *kernel*, juntamente com o Teorema de Mercer e algumas vertentes do SVM, tais como: PSVM e GEPSVM. Em cada uma das situações procuramos exemplificar graficamente, por meio de um conjunto de amostras fictícias, o comportamento de cada algoritmo.

## 3.2 SVM

As Máquinas de Vetores de Suporte, proveniente do Inglês *Support Vector Machine* (SVM), abreviatura que iremos utilizar ao longo do nosso estudo, foram introduzidas por Boser et al. [13] em 1992 e Cortes e Vapnik [28] em 1995. O SVM é um algoritmo que trabalha baseado no princípio da teoria do aprendizado estatístico proposto por Vapnik e Chervonenkis [163] e tem sido muito estudado em todo o mundo, ao longo dos últimos anos. A motivação para considerar as SVMs como classificadores binários vem de limites teóricos sobre a generalização de erro [165]. Para Hearst et al. [58], estudar as SVMs é algo muito útil em dois aspectos: primeiro porque é bem fundamentado do ponto de vista teórico e, segundo, por obter alto desempenho em aplicações práticas.

Na construção das SVMs, queremos minimizar a função convexa (3-1) conforme apresentado a seguir:

$$\min_{\mathbf{w}} \frac{\| \mathbf{w} \|^2}{2}, \tag{3-1}$$

com o intuito de determinar um hiperplano que separa ao máximo as amostras de treinamento, de acordo com a classe que cada uma pertença. A definição desse hiperplano

é apresentada a seguir:

**Definição 3.2.1 (Hiperplano)** *Dados $\boldsymbol{w} \in \mathbb{R}^n, \boldsymbol{w} \neq 0$ e $b \in \mathbb{R}$, o hiperplano $H$ é um conjunto da forma $H = \{\boldsymbol{x} \in \mathbb{R}^n | \boldsymbol{w}^T \boldsymbol{x} = b\}$.*

Sobre a Definição 3.2.1, quando $b = 0$, o hiperplano é simplesmente o conjunto de pontos que são ortogonais a $\mathbf{x}$ e quando $b \neq 0$, o hiperplano é uma translação, ao longo da direção de $\mathbf{x}$, desse conjunto.

A fim de resolver o problema de otimização (3-1), será realizada uma transformação nesse problema, denominado problema primal, em um outro problema, denominado dual [9]. Esse processo é realizado por meio da função Lagrangiana, apresentada a seguir:

**Definição 3.2.2 (Função Lagrangiana)** *[9] Dado um problema de otimização com função objetivo $f(\boldsymbol{w})$ e restrições de igualdade $h_i = 0, i = 1, \ldots, m$. Definimos a função Lagrangiana como:*

$$L(\boldsymbol{w}, \alpha) = f(w) + \sum_{i=1}^{m} \alpha_i h_i(\boldsymbol{w}), \tag{3-2}$$

*em que os coeficientes $\alpha_i$ são chamados multiplicadores de Lagrange.*

A solução do problema de otimização (3-1) é determinada pelo ponto de sela da função Lagrangiana (3-2), que deverá ser minimizada em relação a $\mathbf{w}$ e $b$ e maximizada em relação aos multiplicadores de Lagrange ($\alpha$) [13], para fazer isso, utilizaremos a proposição a seguir:

**Proposição 3.2.3** *[9] Uma condição necessária para um ponto $\boldsymbol{w}^*$ ser um mínimo de $f(\boldsymbol{w})$, sujeito à $h_i = 0, i = 1, \ldots, m$, com $f$, $h_i \in C^1$ (classe $C^1$) é*

$$\frac{\partial L(\boldsymbol{w}^*, \alpha^*)}{\partial \boldsymbol{w}} = 0 \tag{3-3}$$

*e*

$$\frac{\partial L(\boldsymbol{w}^*, \alpha^*)}{\partial \alpha} = 0, \tag{3-4}$$

*para valores $\alpha^*$.*

A equação (3-3) fornece um novo sistema de equações, em que o vetor $\mathbf{w}^*$ é definido em termos de uma expansão que envolve os $N$ exemplos de treinamento [56], e é esse vetor que especifica o hiperplano com margem máxima [13]. A equação (3-4) retorna as restrições de igualdades, resolvendo os dois sistemas, obtém-se a solução.

Quando o problema primal é convexo, as condições de Karush-Kuhn-Tucker (KKT) são suficientes para que o ponto encontrado seja ótimo para o problema primal

e dual [14]. Segundo Bertsekas [9], um vetor $\mathbf{x}^*$ é denominado *regular* se o gradiente $\nabla h_i(x)$ com $i = 1,\ldots,m$, das restrições de igualdade e o gradiente $\nabla g_j(x)$ com $j \in A(X)$, das restrições de desigualdade ativas são linearmente independentes.

Para qualquer ponto factível $x$, o conjunto de restrições de desigualdade ativa é denotado por $A(x) = \{j \mid g_j(x) = 0\}$.

**Proposição 3.2.4 (KKT)** *[9] Seja $x^*$ um mínimo local do problema a seguir:*

$$\begin{aligned} &min \quad f(x), \\ &s.a. \begin{cases} h_1(x) = 0, \ldots, h_m(x) = 0 \\ g_1(x) \leq 0, \ldots, g_r(x) \leq 0 \end{cases}, \end{aligned} \tag{3-5}$$

*em que $f$, $h_i$, $g_j$ são funções continuamente diferenciáveis de $\mathbb{R}^n$ em $\mathbb{R}$. Assuma que $x^*$ é regular. Então existem únicos vetores multiplicadores de Lagrange $\sigma^* = (\sigma_1^*, \ldots, \sigma_m^*)$, $\mu^* = (\mu_1^*, \ldots, \mu_r^*)$, tais que:*

$$\begin{aligned} \nabla_x L(x^*, \sigma^*, \mu^*) &= 0, \\ \mu_j^* \geq 0, \quad j &= 1, \ldots, r, \\ \mu_j^* = 0, \quad \forall j &\notin A(x^*), \end{aligned} \tag{3-6}$$

*em que $A(x^*)$ é o conjunto de restrições ativas de $x^*$.*

Para o caso em que os dados não são linearmente separáveis, é necessário mapear esses dados para um espaço de alta dimensão. Esse espaço é denominado espaço de característica, definido por meio de uma função $\phi$, denominada *função kernel*, que recebe dois pontos $\mathbf{x}_i$ e $\mathbf{x}_j$ do espaço de entrada e calcula o produto interno $\phi(\mathbf{x}_i)^T \phi(\mathbf{x}_j)$ no espaço de alta dimensão [94]. A escolha apropriada dessa função faz com que o conjunto de treinamento mapeado para o espaço de característica seja separado por um hiperplano linear [95]. Em 1909, Mercer provou um teorema que define uma forma geral de produtos internos em espaços de Hilbert, este teorema será apresentado no Capítulo 3 [164]. A seguir são definidos o Espaço de Hilbert, produto interno e a definição de compacidade, relativas ao Teorema de Mercer.

**Definição 3.2.5 (Espaço de Hilbert)** *[88] Um espaço de Hilbert é um espaço vetorial $H$, munido de um produto interno, e completo em relação a norma definida por esse produto interno.*

**Definição 3.2.6 (Produto Interno)** *[88] Seja $E$ um espaço vetorial real. Um produto interno em $E$ é uma função $\langle , \rangle : E \times E \to \mathbb{R}$, que associa a cada par ordenado de vetores*

*$\boldsymbol{x},\boldsymbol{y} \in E$ um número real $\langle \boldsymbol{x},\boldsymbol{y} \rangle$, que denotaremos ao longo do nosso estudo por $\boldsymbol{x}^T\boldsymbol{y}$ , chamado o produto interno de $\boldsymbol{x}$ por $\boldsymbol{y}$, de modo a serem cumpridas as condições a seguir, para $\boldsymbol{x}$, $\boldsymbol{y}$ e $\boldsymbol{z} \in E$ e $\lambda \in \mathbb{R}$.*

**P1)** $(\boldsymbol{x}+z)^T\boldsymbol{y} = \boldsymbol{x}^T\boldsymbol{y} + z^T y$;
**P2)** $(\lambda\boldsymbol{x})^T\boldsymbol{y} = \lambda \cdot \boldsymbol{x}^T\boldsymbol{y}$;
**P3)** $\boldsymbol{x}^T\boldsymbol{y} = \boldsymbol{y}^T\boldsymbol{x}$;
**P4)** $x \neq 0 \Rightarrow \boldsymbol{x}^T\boldsymbol{x} > 0$.

**Teorema 3.2.7** *[89] As seguintes afirmações a respeito de um conjunto $K \subset \mathbb{R}$ são equivalentes:*

1. *K é limitado e fechado;*
2. *toda cobertura aberta de K possui subcobertura finita;*
3. *todo subconjunto infinito de K possui ponto de acumulação pertencente a K;*
4. *toda sequência de ponto de K possui uma subsequência que converge para um ponto de K.*

**Definição 3.2.8 (Conjunto Compacto)** *[89] Um conjunto $K \subset \mathbb{R}$, é denominado compacto se cumpre uma (e portanto todas as) condições do Teorema 3.2.7.*

Em algumas situações, se faz necessário a escolha de uma norma apropriada. Zhu et al. [195] relatam que SVM utilizando norma-2 é bem conhecida pelo seu bom desempenho. Já Gao et al. [44] propuseram a mudança da norma-2 para norma-1, mostrando que para o caso proposto, essa foi melhor. A seguir é apresentada a definição de norma, seguida de algumas normas muito utilizadas no contexto que iremos trabalhar.

**Definição 3.2.9 (Norma)** *[88] Seja E um espaço vetorial real. Uma norma em E é uma função real $\| \quad \|: E \to \mathbb{R}$, que associa a cada vetor $\boldsymbol{x} \in E$ o número real $| \boldsymbol{x} |$, chamado a norma de $\boldsymbol{x}$, de modo a serem cumpridas as condições abaixo para quaisquer $\boldsymbol{x},\boldsymbol{y} \in E$ e $\lambda$ escalar:*

1. *Se $\boldsymbol{x} \neq 0$ então $\| \boldsymbol{x} \| \neq 0$;*
2. $\| \lambda \cdot \boldsymbol{x} \| = |\lambda| \cdot \| \boldsymbol{x} \|$;
3. $\| \boldsymbol{x} + \boldsymbol{y} \| \leq \| \boldsymbol{x} \| + \| \boldsymbol{y} \|$ .

Dado um vetor do $\mathbb{R}^n$, seguem algumas normas importantes:

- Norma-2 ou norma Euclideana: $\| \mathbf{x} \|_2 = \sqrt{\mathbf{x}^T\mathbf{x}} = \sqrt{\langle \mathbf{x},\mathbf{x} \rangle} = \left( \sum_{i=1}^{n} | \mathbf{x}_i |^2 \right)^{\frac{1}{2}}$;
- Norma-1: $\| \mathbf{x} \|_1 = \sum_{i=1}^{n} | \mathbf{x}_i |$;

- Norma-$r$ : $\| \mathbf{x} \|_r = \left( \sum_{i=1}^{n} | \mathbf{x}_i |^r \right)^{\frac{1}{r}}, \quad 1 \leq r < \infty;$
- Norma-$\infty$ : $\| \mathbf{x} \|_\infty = \max_{1 \leq i \leq n} | \mathbf{x}_i |$ .

Neste capítulo iremos ilustrar graficamente o comportamento de cada algoritmo de classificação com um banco de dados fictício, para facilitar a abstração, conforme ilustra a Fig. (3.1). Descreveremos a seguir a fundamentação teórica sobre o SVM linear de margens rígidas e suaves e o SVM não linear.

### 3.2.1 SVM lineares com margens rígidas

Dado um conjunto de pares de treinamento, de amostras e rótulos, denotados por $(\mathbf{x}_i, y_i)$, com $\mathbf{x}_i \in \mathbb{R}^n$ e $y \in [-1, 1]$, a margem de um hiperplano representa a soma das distâncias entre o hiperplano e os dados positivos mais próximos e, entre o hiperplano e os dados negativos mais próximos. A Fig. (3.1a) apresenta um conjunto de dados linearmente separáveis, a Fig. (3.1b) apresenta os possíveis hiperplanos separadores entre os conjuntos de dados e a Fig. (3.1c) apresenta o hiperplano separador de margem máxima.

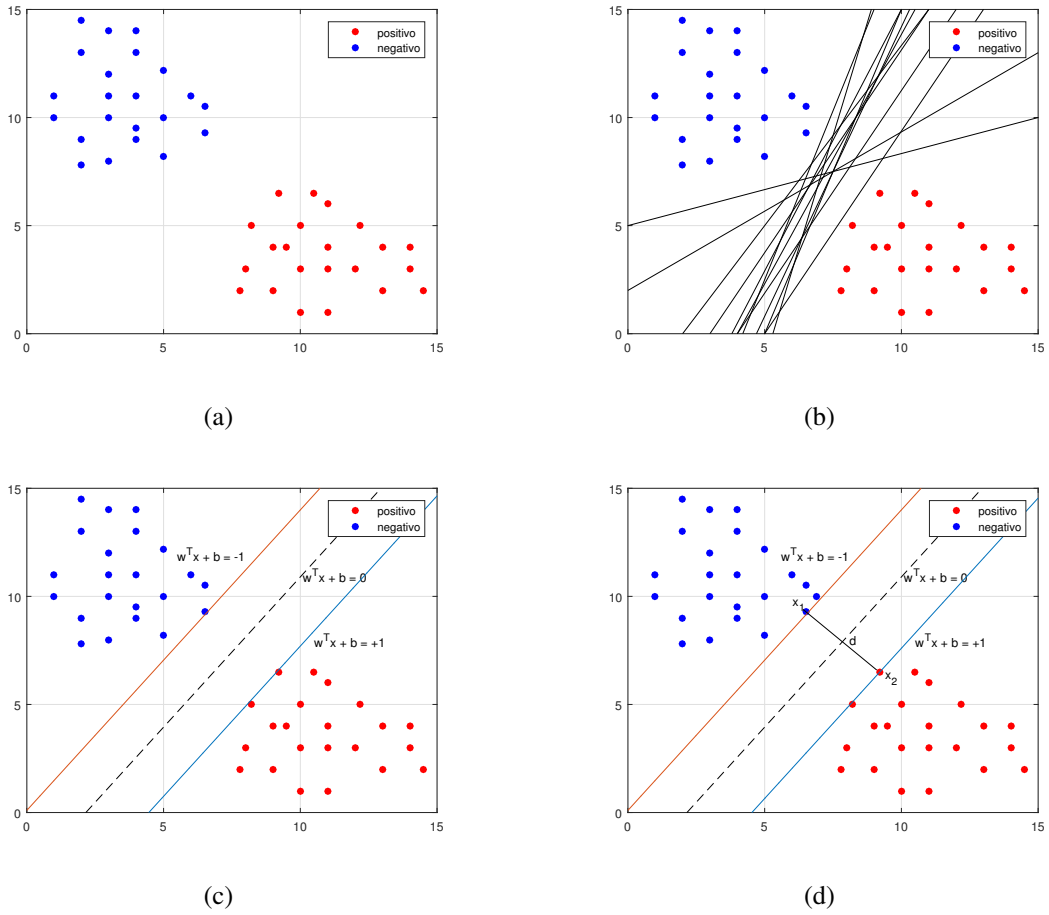



**Fig 3.1:** *Gráfico com dados fictícios sendo: (a) amostras no plano $\mathbb{R}^2$, (b) alguns possíveis hiperplanos separadores, (c) hiperplano separador com margem máxima, (d) distância entre os hiperplanos de margem máxima.*

Note que não existem amostras entre os hiperplanos $\mathbf{w}^T x + b = -1$ e $\mathbf{w}^T x + b = +1$ na Fig. (3.1c), deste modo, as amostras são classificadas pelas inequações em (3-7):

$$\begin{cases} \mathbf{w}^T \mathbf{x}_i + b \geq +1 \quad \text{se} \quad y_i = +1 \\ \mathbf{w}^T \mathbf{x}_i + b \leq -1 \quad \text{se} \quad y_i = -1 \\ \qquad i = 1, 2, \ldots, n \end{cases} . \tag{3-7}$$

Essas duas inequações podem ser combinadas em uma só, apresentada a seguir:

$$\left\{ y_i(\mathbf{w}^T \mathbf{x}_i + b) - 1 \geq 0 \qquad i = 1, 2, \ldots, n. \right. \tag{3-8}$$

O hiperplano ótimo é ortogonal ao menor segmento conectando os hiperplanos

$\mathbf{w}^T\mathbf{x}_i + b = -1$ e $\mathbf{w}^T\mathbf{x}_i + b = +1$. Considerando que $\mathbf{x}_1$ e $\mathbf{x}_2$ estão, respectivamente, sobre estes hiperplanos, e que são os pontos de menor distância entre os hiperplanos, então a distância entre $\mathbf{x}_1$ e $\mathbf{x}_2$ é dada pela Eq. (3-9) a seguir:

$$\mathbf{w}^T(\mathbf{x}_2 - \mathbf{x}_1) = 2, \tag{3-9}$$

em que $\mathbf{w}$ e $(\mathbf{x}_2 - \mathbf{x}_1)$ são paralelos entre si e ortogonais ao hiperplano que separa os conjuntos de amostras conforme Fig. (3.1d). Temos que:

$$\mid \mathbf{w}^T(\mathbf{x}_2 - \mathbf{x}_1) \mid = \parallel \mathbf{w} \parallel \times \parallel \mathbf{x}_2 - \mathbf{x}_1 \parallel, \tag{3-10}$$

em que $\parallel \cdot \parallel$ é a norma vetorial. Das Equações (3-9) e (3-10) temos:

$$\parallel \mathbf{x}_2 - \mathbf{x}_1 \parallel = \frac{2}{\parallel \mathbf{w} \parallel}. \tag{3-11}$$

Como queremos maximizar a margem que separa o conjunto de amostras, teremos então que minimizar $\parallel \mathbf{w} \parallel$, ou seja, temos um problema de otimização, denominado problema primal que é minimizar $\frac{\|\mathbf{w}\|}{2}$, sujeito às restrições apresentadas pela inequação (3-8), conforme a seguir:

$$\begin{aligned} &\min_{\mathbf{w}} \frac{\parallel \mathbf{w} \parallel^2}{2}, \\ &\text{s.a.} \left\{ y_i(\mathbf{w}^T\mathbf{x}_i + b) - 1 \geq 0. \right. \end{aligned} \tag{3-12}$$

Para resolvermos (3-12), introduziremos a função Lagrangiana, definida em função de $\mathbf{w}$ e $b$, conforme sugere Cristianini et al. [30], assim a tarefa de aprendizagem se reduz à minimização da função objetivo:

$$L(\mathbf{w}, b, \alpha) = \frac{1}{2} \parallel \mathbf{w} \parallel^2 - \sum_{i=1}^{n} \alpha_i (y_i(\mathbf{w}^T\mathbf{x}_i + b) - 1), \tag{3-13}$$

em que $\alpha_i$ são denominados multiplicadores de Lagrange.

Segundo Haykin [56], a solução para o problema de otimização restrito é determinada pelo ponto de sela da função Lagrangiana (3-13), que deverá ser minimizada em relação a $\mathbf{w}$ e $b$ e deve ser maximizada em relação a $\alpha$.

Para obter os pontos ótimos de (3-13), basta diferenciarmos $L(\mathbf{w}, b, \alpha)$ em relação a $\mathbf{w}$ e a $b$ e igualar os resultados a zero, conforme a seguir:

$$\frac{\partial L(\mathbf{w}, b, \alpha)}{\partial \mathbf{w}} = 0 \tag{3-14}$$

e

$$\frac{\partial L(\mathbf{w}, b, \alpha)}{\partial b} = 0. \tag{3-15}$$

Desenvolvendo um pouco mais (3-13) temos o seguinte:

$$L(\mathbf{w}, b, \alpha) = \frac{1}{2} \parallel \mathbf{w} \parallel^2 - \sum_{i=1}^{n} \alpha_i y_i \mathbf{w}^T \mathbf{x}_i - \sum_{i=1}^{n} \alpha_i y_i b + \sum_{i=1}^{n} \alpha_i. \tag{3-16}$$

Logo,

$$\frac{\partial L(\mathbf{w}, b, \alpha)}{\partial \mathbf{w}} = 0 \Rightarrow \mathbf{w} - \sum_{i=1}^{n} \alpha_i y_i \mathbf{x}_i = 0 \Rightarrow \mathbf{w} = \sum_{i=1}^{n} \alpha_i y_i \mathbf{x}_i \tag{3-17}$$

e

$$\frac{\partial L(\mathbf{w}, b, \alpha)}{\partial b} = 0 \Rightarrow - \sum_{i=1}^{n} \alpha_i y_i = 0 \Rightarrow \sum_{i=1}^{n} \alpha_i y_i = 0. \tag{3-18}$$

Obtendo os resultados representados pelas Equações (3-19) e (3-20) a seguir:

$$\mathbf{w} = \sum_{i=1}^{n} \alpha_i y_i \mathbf{x}_i \tag{3-19}$$

e

$$\sum_{i=1}^{n} \alpha_i y_i = 0. \tag{3-20}$$

O problema primal trabalha com uma função de custo convexa e com restrições lineares. Assim, dado um problema de otimização restrito como esse, é possível construir um outro problema denominado problema dual [56], que tem o mesmo valor ótimo do problema primal, mas com os multiplicadores de Lagrange fornecendo a solução ótima. Segundo Bertsekas [9], se o problema primal tem uma solução ótima, então o problema dual também tem uma solução ótima, e os valores ótimos correspondentes são iguais, e para que $\mathbf{w}$ seja uma solução primal ótima e $\alpha$ seja uma solução dual ótima, é necessário e suficiente que $\mathbf{w}$ seja solução para o problema primal.

Fazendo a substituição das equações (3-19) e (3-20) na Lagrangiana (3-13), e nomeando a função objetivo do problema dual por $L_D(\alpha)$, obtemos o seguinte:

$$L_D(\alpha) = \frac{1}{2} \sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j \mathbf{x}_i^T \mathbf{x}_j - \sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j \mathbf{x}_i^T \mathbf{x}_j + \sum_{i=1}^{n} \alpha_i, \tag{3-21}$$

agrupando temos:

$$L_D(\alpha) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j \mathbf{x}_i^T \mathbf{x}_j. \tag{3-22}$$

Assim, o problema dual é expresso em (3-23):

$$\begin{aligned} &\max_{\alpha} \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j \mathbf{x}_i^T \mathbf{x}_j, \\ &\text{s.a.} \begin{cases} \alpha_i \geq 0, i = 0, 1, \ldots, n \\ \sum_{i=1}^{n} \alpha_i y_i = 0 \end{cases}, \end{aligned} \tag{3-23}$$

onde obtemos uma solução ótima para $\alpha$ que designaremos por $\overline{\alpha}$, assim podemos substituí-lo na Eq. (3-19) para obtermos o valor ótimo para $\mathbf{w}$ que denotaremos por $\overline{\mathbf{w}}$, o valor ótimo para $b$ que denotaremos por $\overline{b}$ pode ser calculado pela expressão a seguir:

$$\overline{b} = -\frac{1}{2} \left[ \max_{\{i|y_i=-1\}} \left( \sum_{j=1}^{n} y_j \overline{\alpha}_j \mathbf{x}_i^T \mathbf{x}_j \right) + \min_{\{i|y_i=+1\}} \left( \sum_{j=1}^{n} y_j \overline{\alpha}_j \mathbf{x}_i^T \mathbf{x}_j \right) \right], \tag{3-24}$$

dessa forma chegamos ao classificador que denotaremos por $g(x)$, esta função linear representa o hiperplano que separa os dados com margem máxima,

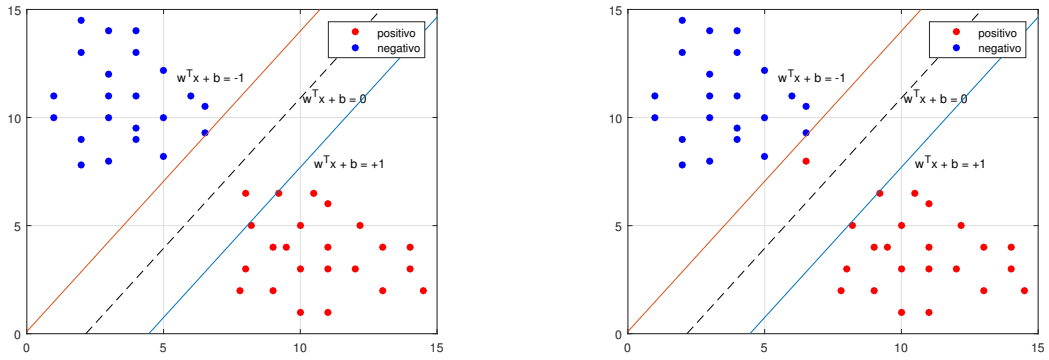
$$g(x) = sgn\left( \sum_{\mathbf{x}_i \in SV} \overline{\alpha}_i y_i \mathbf{x}_i^T \mathbf{x} + \overline{b} \right), \tag{3-25}$$

onde uma amostra será classificada como positiva se $\sum_{\mathbf{x}_i \in SV} \overline{\alpha}_i y_i \mathbf{x}_i^T \mathbf{x} + \overline{b} > 0$ e será classificada como negativa se $\sum_{\mathbf{x}_i \in SV} \overline{\alpha}_i y_i \mathbf{x}_i^T \mathbf{x} + \overline{b} < 0$, onde $SV$ são os vetores de suportes.

### 3.2.2 SVM lineares com margens suaves

Em aplicações com dados reais, não temos a garantia de que os conjuntos de dados se apresentem de modo linearmente separáveis, pois a presença de ruídos ou até mesmo a natureza do problema são alguns obstáculos [16, 95].

As Figs. (3.2a-b) apresentam o caso em que a amostra está entre a margem de separação. Na Fig. (3.2a), a amostra está do lado correto da superfície de decisão, porém dentro da margem e na Fig. (3.2b) a amostra se encontra do outro lado da superfície de decisão.




(a) (b)

**Fig 3.2:** *Gráfico com dados fictícios sendo: (a) com uma amostra entre as margens, mas dentro da superfície de decisão, (b) com uma amostra entre as margens, mas do lado errado da superfície de decisão.*

Para resolver esse problema, conforme sugere Cortes e Vapnik [28], iremos introduzir um conjunto de variáveis escalares não negativas, denominadas variáveis de folga, que denotaremos por $\xi$. Essas variáveis atuarão como penalização para a violação das restrições do problema (3-12). Desse modo, se uma amostra violar a restrição, estando a uma distância menor do que 1, então haverá um aumento na função objetivo do problema primal,

$$\begin{cases} \mathbf{w}^T\mathbf{x}_i + b \geq +1 - \xi_i & \text{se} \quad y_i = +1 \\ \mathbf{w}^T\mathbf{x}_i + b \leq -1 + \xi_i & \text{se} \quad y_i = -1 \quad \text{com} \quad i = 1,2,\ldots,n. \\ \xi_i \geq 0 \end{cases} \tag{3-26}$$

Essas duas inequações, de modo análogo ao feito para o caso linearmente separável, podem ser combinadas em uma só, assim a formulação, para o problema primal, para o caso não linearmente separável é apresentado como segue:

$$\begin{aligned} &\min_{w,\xi_i} \frac{\|\mathbf{w}\|^2}{2} + C\left(\sum_{i=1}^{n} \xi_i\right), \\ &\text{s.a.} \begin{cases} y_i(\mathbf{w}^T\mathbf{x}_i + b) \geq 1 - \xi_i \\ \xi_i \geq 0 \end{cases} \quad \text{com} \quad i = 1,2,\ldots,n, \end{aligned} \tag{3-27}$$

em que $C$ é um parâmetro positivo especificado pelo usuário.

De modo análogo ao realizado na Seção (3.2.1), utilizamos os multiplicadores

de Lagrange $\alpha$ e $\beta$ para determinar o Lagrangiano dual:

$$L_D(\alpha,\beta) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j \mathbf{x}_i^T \mathbf{x}_j, \tag{3-28}$$

embora essa equação se assemelhe ao Lagrangiano dual para dados linearmente separáveis, as restrições impostas sobre os multiplicadores de Lagrange $\alpha_i$ são diferentes,

$$\begin{aligned} &\max_{\alpha} \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j \mathbf{x}_i^T \mathbf{x}_j, \\ &\text{s.a.} \begin{cases} 0 \leq \alpha_i \leq C, \quad \forall i = 1,2,\ldots,n \\ \sum_{i=1}^{n} \alpha_i y_i = 0 \end{cases} . \end{aligned} \tag{3-29}$$

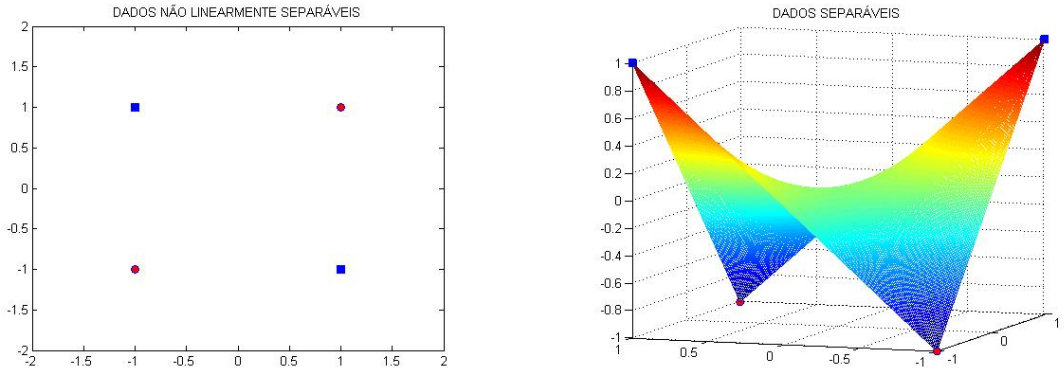
Note que os dados de treinamento aparecem como produto interno e as variáveis de folgas e o multiplicador de Lagrange $\beta$ não aparacem no problema (3-29), o que reduz o número de variáveis envolvidas. Para uma leitura mais detalhada sobre esse tópico, sugerimos Cristianini e Shawe-Taylor [30] e Haykin [56].

### 3.2.3 SVM não lineares

Para problemas reais, as amostras se apresentam bem misturadas, não sendo possível, sem técnicas mais aperfeiçoadas, determinar um hiperplano ótimo de separação. Nesse caso, para conseguirmos essa separação por um hiperplano ótimo, as amostras de treinamento são mapeadas do espaço de origem para um espaço de Hilbert $H$, por meio da função $\Phi$, denominada função *kernel*. Esse mapeamento faz que com seja possível encontrar um hiperplano que separa bem as amostras. Seja $\Phi : \mathbb{R}^d \rightarrow H$, onde $\mathbb{R}^d$ é o espaço de Hilbert com os dados iniciais, ou seja, o espaço original. $H$ é o espaço de Hilbert para onde foram mapeados os dados por meio da função $\Phi$, denominado espaço de características [56]. A motivação, para realização desta aplicação, é dada pelo Teorema de Cover [29], onde um mapeamento não linear é realizado para transformar dados de um problema não linearmente separáveis em um problema linearmente separável.

Uma ideia gráfica do que pretendemos com a função *kernel* é apresentado nas Fig. (3.3). A Fig. (3.3a) apresenta o gráfico cujas amostras $\{(-1,-1),(1,1),(-1,1),(1,-1)\}$ são rotuladas por $\{-1,-1,1,1\}$, respectivamente. Note que neste caso não é possível separar essas amostras por um hiperplano, considerando que, após a aplicação da função *kernel* ($\Phi : \mathbb{R}^2 \rightarrow \mathbb{R}^3$), tenhamos a situação

apresentada na Fig. (3.3b), note que agora é possível separar os dados por um hiperplano, conforme o caso linearmente separável.




(a) (b)

**Fig 3.3:** *Gráfico das amostras* $(-1,-1)$, $(-1,+1)$, $(+1,-1)$ *e* $(+1,+1)$ *com rótulos* $\{-1,+1,+1,-1\}$, *respectivamente, sendo (a) não linearmente separável, (b) linearmente separável.*

O hiperplano ótimo é obtido utilizando as equações apresentadas na Seção 3.2.2, onde a função $\Phi$ é aplicada nas amostras da função objetivo do problema de otimização (3-29), representada conforme a seguir:

$$\max_{\alpha} \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j \Phi^T(\mathbf{x}_i)\Phi(\mathbf{x}_j). \tag{3-30}$$

Uma das vantagens da aplicação da função *kernel* é que precisamos realizar somente o cálculo de produtos escalares entre os dados no espaço de alta dimensão, denotado no espaço de característica por $\Phi^T(\mathbf{x}_i)\Phi(\mathbf{x}_j)$, que dependem apenas de $\mathbf{x}_i$ e $\mathbf{x}_j$ pertencentes ao espaço de origem. Mas calcular $\Phi$ explicitamente, não é uma tarefa fácil, conforme cita Cristianini [30], pois $H$ pode ter uma dimensão muito grande, podendo até ser infinita. Para solucionar esse problema, iremos apresentar a função *kernel*, conforme a seguir.

### 3.2.4 Funções *Kernel*

Um *kernel* $K$ é uma função que recebe dois valores $\mathbf{x}_i$ e $\mathbf{x}_j$ do espaço de origem e calcula o produto escalar $\Phi^T(\mathbf{x}_i)\Phi(\mathbf{x}_j)$ no espaço de característica, conforme definição a seguir:

**Definição 3.2.10** *[138] Seja $X$ um conjunto não vazio. A função $K : X \times X \to \mathbb{R}$ é dita ser um kernel, se existe um espaço de Hilbert $H$ e uma aplicação $\Phi : X \to H$, tal que para*

*todo* $\boldsymbol{x},\boldsymbol{y} \in H$,

$$K(\boldsymbol{x},\boldsymbol{y}) = \Phi^T(\boldsymbol{x})\Phi(\boldsymbol{y}). \tag{3-31}$$

A teoria sobre a função *kernel* utiliza das condições do Teorema de Mercer [103], conforme apresentado a seguir:

**Teorema 3.2.11 (Teorema de Mercer)** *[103] Seja $K$ um kernel simétrico, contínuo semi-definido positivo sobre um conjunto compacto $X$, e o operador integral $T_k : L_2(X) \to L_2(X)$ definido por*

$$(T_k f)(\cdot) = \int_X K(\cdot,x) f(x) d(x)$$

*é semi-definido positivo, ou seja, para todo $f \in L_2(X)$,*

$$\int_X K(u,v) f(u) f(v) du dv \geq 0,$$

*então há uma base ortonormal $\{\psi_i\}$ de $L_2(x)$ consistindo de autofunções de $T_K$, tal que a correspondente sequência de autovalores $\{\lambda_i\}$ são não-negativos. As autofunções correspondentes para autovalores não nulos são contínuas sobre $X$ e $K(u,v)$ tem a seguinte representação:*

$$K(u,v) = \sum_{i=1}^{\infty} \lambda_i \psi_i(u) \psi_i(v),$$

*onde a convergência é absoluta e uniforme, ou seja;*

$$\lim_{n\to\infty} \sup_{u,v} \mid K(u,v) - \sum_{i=1}^{n} \lambda_i \psi_i(u) \psi_i(v) \mid = 0.$$

Em decorrência do Teorema de Mercer, dado um algoritmo que é escrito em função de um *kernel* positivo definido $K$, pode se construir uma versão alternativa escrita em função de outro *kernel* positivo definido $\tilde{K}$ [102, 137]. Essa mudança no *kernel* é conhecida como o truque do *kernel* (*Kernel Trick*).

**Exemplo 3.2.12 (Kernel)** *Existem vários kernels disponíveis na literatura, mas os mais utilizados são:*

- *kernel Polinomial*

$$K(\boldsymbol{x},\boldsymbol{y}) = (\boldsymbol{x}^T\boldsymbol{y} + \theta)^d, \quad \textit{com } d \in \mathbb{N} \textit{ e } \theta \textit{ definidos pelo usuário.}$$

- *kernel Gaussiano*

$$K(\boldsymbol{x},\boldsymbol{y}) = e^{-\frac{1}{2\sigma^2}\|\boldsymbol{x}-\boldsymbol{y}\|^2}, \quad \textit{com } \sigma^2 \textit{ definido pelo usuário.}$$

- *kernel Sigmoide ou Tangente Hiperbólico*

$$K(\boldsymbol{x},\boldsymbol{y}) = tanh(\eta\boldsymbol{xy} + \theta), \quad com\ \eta,\theta \in \mathbb{R}.$$

Nesse último caso, o Teorema de Mercer é satisfeito somente para alguns valores de $\eta$ e $\theta$, pois não é positivo definido. Para uma leitura mais aprofundada sobre *kernel* sugerimos Cristianini e Shawe-Taylor [30], Evgeniou et al. [41], Haykin [56] e Scholkopf e Smola [137].

## 3.3 Máquinas de Vetores de Suporte Proximal

Máquinas de vetores de suporte proximal (*Proximal Support Vector Machine) PSVM* foi proposto por Fung e Mangasarian [43], com o objetivo de classificar amostras dependendo da proximidade de um dos dois planos paralelos e que estejam o mais distante possível do outro. O PSVM pode ser interpretado como uma solução do sistema de mínimos quadrados regularizado [69].

### 3.3.1 PSVM Linear

A diferença entre o SVM e PSVM está na formulação, e consequentemente no seu comportamento, onde a primeira das desigualdades das restrições do problema (3-27) é trocada pela igualdade, e não é necessária a imposição sobre a segunda restrição. Na função objetivo, é adicionada o termo $b^2$ a $\mathbf{w}^T\mathbf{w}$ ajudando na maximização da margem entre os hiperplanos paralelos de separação, conforme sugerem Mangasarian e Musicant [100]. É feito também a mudança da norma-1 para norma-2 da variável de folga. A seguir é apresentado o problema de otimização:

$$\begin{aligned} \min_{\mathbf{w},b,\xi} \quad & \frac{1}{2}(\mathbf{w}^T\mathbf{w} + b^2) + \frac{C}{2} \parallel \xi_i \parallel^2, \\ \text{s.a.} \quad & \Big\{ D(A\mathbf{w} - \mathbf{e}b) + \xi = \mathbf{e} \quad , \end{aligned} \tag{3-32}$$

em que $A$ é um matriz $(m \times n)$ e $D$ uma matriz diagonal $(m \times m)$ composta por 1 e $-1$. De fato, ela retorna o que podemos chamar de solução exata. Desse modo, os planos $\mathbf{w}^T\mathbf{x} + b = \pm 1$ não são limitantes, mas podem ser pensados como planos próximos a cada agrupamento de pontos e que são empurrados o mais distante possível pelo termo $(\mathbf{w}^T\mathbf{w} + b^2)$ na função objetivo, que não é nada além do que o inverso da distância com norma-2 entre os dois planos no $(\mathbf{w},b)$-espaço de $\mathbb{R}^{n+1}$. Algumas aplicações podem ser

encontradas em Fung e Mangasarian [43] e Khemchandani et al. [69]. A Fig.(3.4) mostra os hiperplanos paralelos, cada um mais próximo ao seu agrupamento de amostras.

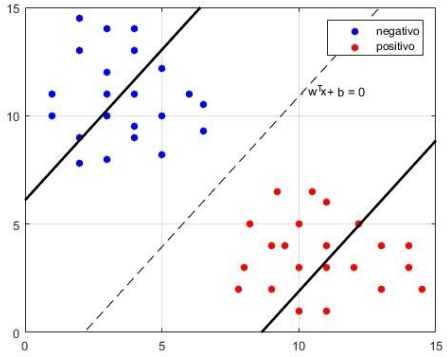



**Fig 3.4:** *Gráfico PSVM linearmente separável.*

Omitiremos a descrição do caso não linear do PSVM, para não fugir do foco do trabalho, podendo ser encontrada em Fung e Mangasarian [43].

## 3.4 Máquinas de Vetores de Suporte Proximal por Generalização de Autovalores

As máquinas de vetores de suporte proximal por generalização de autovalores *Generalized Eigenvalues Proximal Support Vector Machine* (GEPSVM), foram propostas por Mangasarian e Wild [101]. Diferentemente do SVM (Seção 3.2) e PSVM (Seção 3.3), em que os dois hiperplanos procurados são paralelos, a formulação do GEPSVM não permanece com a condição de paralelismo entre os hiperplanos, mas exige que cada hiperplano esteja o mais próximo possível a um agrupamento de dados e o mais longe possível do outro agrupamento de dados e vice-versa. Sejam os dois hiperplanos no $\mathbb{R}^n$ expressos por:

$$\mathbf{w}^T\mathbf{x}_1 + b_1 = 0 \quad \text{e} \quad \mathbf{w}^T\mathbf{x}_2 + b_2 = 0. \tag{3-33}$$

Para obter o primeiro hiperplano, Mangasarian e Wild [101] minimizaram a soma do quadrado das distâncias entre cada amostra de uma das classes ao plano, utilizando norma-2, e dividiram o resultado pelo quadrado das distâncias entre cada amostra da outra classe ao plano, também utilizando norma-2, conforme o problema de otimização

que segue:

$$\min_{(\mathbf{w},b)\neq 0} \frac{\| A\mathbf{w}-\mathbf{e}b \|^2 / \| \begin{bmatrix} \mathbf{w} \\ b \end{bmatrix} \|^2}{\| B\mathbf{w}-\mathbf{e}b \|^2 / \| \begin{bmatrix} \mathbf{w} \\ b \end{bmatrix} \|^2}. \tag{3-34}$$

em que $A$ é uma matriz $(m_1 \times n)$ pertencente à classe 1, $B$ é uma matriz $(m_2 \times n)$ pertencente à classe $-1$, com $(m_1 + m_2 = m)$, sendo $m$ o número de amostras e $n$ o número de variáveis.

Para o outro hiperplano, o processo de cálculo é análogo, bastando inverter a fração do problema (3-34), e para ambos os casos foi necessário a introdução do termo de regularização de Tikhonov [153], que é muito usado para mínimos quadrados regularizados, e reduz a norma das variáveis (**w**, b) do problema, conforme apresentado a seguir:

$$\min_{(\mathbf{w},b)\neq 0} \frac{\| A\mathbf{w}-\mathbf{e}b \|^2 + \delta \| \begin{bmatrix} \mathbf{w} \\ b \end{bmatrix} \|^2}{\| B\mathbf{w}-\mathbf{e}b \|^2}. \tag{3-35}$$

O GEPSVM produz o seguinte par de problemas de minimização:

$$\min_{(\mathbf{w},b)\neq 0} \frac{\| A\mathbf{w}-\mathbf{e}b \|^2 + \delta \| \begin{bmatrix} \mathbf{w} \\ b \end{bmatrix} \|^2}{\| B\mathbf{w}-\mathbf{e}b \|^2} \quad \text{e} \quad \min_{(\mathbf{w},b)\neq 0} \frac{\| B\mathbf{w}-\mathbf{e}b \|^2 + \delta \| \begin{bmatrix} \mathbf{w} \\ b \end{bmatrix} \|^2}{\| A\mathbf{w}-\mathbf{e}b \|^2} \tag{3-36}$$

em que $\delta$ é um parâmetro não negativo.

Os problemas de otimização (3-36) podem ser reformulados como segue:

$$\min_z \frac{\mathbf{z}^T G\mathbf{z}}{\mathbf{z}^T H\mathbf{z}} \quad \text{e} \quad \min_z \frac{\mathbf{z}^T L\mathbf{z}}{\mathbf{z}^T M\mathbf{z}}, \tag{3-37}$$

em que,

$$G = [A \quad -\mathbf{e}]^T [A \quad -\mathbf{e}] + \delta I, \quad H = [B \quad -\mathbf{e}]^T [B \quad -\mathbf{e}], \quad \mathbf{z} = \begin{bmatrix} \mathbf{w} \\ b \end{bmatrix},$$
$$L = [B \quad -\mathbf{e}]^T [B \quad -\mathbf{e}] + \delta I \quad \text{e} \quad M = [A \quad -\mathbf{e}]^T [A \quad -\mathbf{e}].$$

O problema (3-37) é conhecido como quociente de Rayleigh [113], cuja solução global ótima é obtida resolvendo os seguintes problemas de autovalores generalizados:

$$G\mathbf{z} = \lambda H\mathbf{z} \quad \text{e} \quad L\mathbf{z} = \lambda H\mathbf{z}. \tag{3-38}$$

Os mínimos do problema (3-36) são obtidos pelos menores autovalores do problema de autovalores generalizados (3-38), conforme descrevem Mangasararian e Wild [101].

A Fig. (3.5) mostra o comportamento dos hiperplanos, podemos notar que cada plano está mais próximo a uma classe e o mais longe possível da outra classe de amostras.

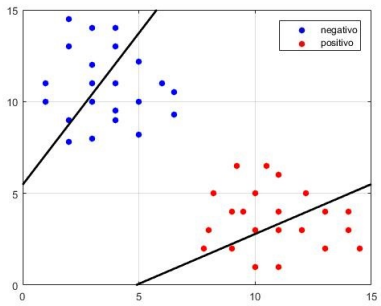



**Fig 3.5:** *Gráfico GEPSVM linearmente separável.*

Algumas análises de desempenho e precisão sobre o GEPSVM, bem como a extensão para o caso não linear e comparação com outros algoritmos, podem ser encontradas em Chen et al. [25] e Mangasarian e Wild [101].

## 3.5 Considerações Finais

Neste capítulo, foi apresentada a estruturação matemática das SVMs, bem como a explanação sobre a função *kernel*, que auxilia na projeção das amostras do espaço de entrada para o espaço de alta dimensão, onde é feito o uso do Teorema de Mercer. São apresentadas também algumas extensões do SVM, tais como: PSVM e GEPSVM. Embora existam várias extensões do SVM na literatura, apresentamos somente a construção dessas duas extensões, com o intuito de criar uma sequência lógica do comportamento, de como os algoritmos foram estruturados em relação à construção dos hiperplanos, e assim apresentarmos a próxima extensão do SVM, que são as máquinas de vetores de suportes duplos (*Twin Support Vector Machine* (TWSVM)) no Capítulo 4.

CAPÍTULO 4

# Máquinas de Vetores de Suporte Duplo (TWSVM) e extensão

## 4.1 Considerações Iniciais

Neste capítulo, são apresentadas as construções teórica do TWSVM para o caso linear e não linear, para problemas de classificação binária e uma de suas extensões, denominada TBSVM.

## 4.2 TWSVM

Máquinas de vetores de suporte duplo, do inglês *twin support vector machine* (TWSVM), foi introduzido por Jayadeva et al. em 2007 [66]. A ideia do TWSVM é construir dois planos não paralelos, onde cada plano é determinado pela resolução de um problema de programação quadrática, ou seja, resolver dois problemas ao invés de um, conforme descrito no Capítulo 3 para o SVM.

A seguir apresentaremos a estruturação matemática do TWSVM para o caso linear e não linear.

## 4.3 TWSVM linear

Considerando um problema de classificação binário, onde os pontos pertencentes às classes 1 e $-1$ são representados pelas matrizes $A \in \mathbb{R}^{m_1 \times n}$ e $B \in \mathbb{R}^{m_2 \times n}$, respectivamente, em que $m_1$ representa a quantidade de padrões da classe $A$ e $m_2$ representa a quantidade de padrões da classe $B$. TWSVM procura um par de hiperplanos não paralelos

$$\mathbf{w}_1^T \mathbf{x} + b_1 = 0 \quad \text{e} \quad \mathbf{w}_2^T \mathbf{x} + b_2 = 0, \tag{4-1}$$

em que $\mathbf{w}_1, \mathbf{w}_2 \in \mathbb{R}^n$, $b_1, b_2 \in \mathbb{R}$, e cada hiperplano esteja o mais próximo do conjunto de pontos de uma das classes e o mais distante da outra classe e vice-versa. A proposta

do TWSVM linear é resolver o par de problemas de programação quadrática, conforme apresentado a seguir:

$$\begin{aligned} &\min_{\mathbf{w}_1,b_1,\xi_2} \frac{1}{2}(A\mathbf{w}_1+\mathbf{e}_1b_1)^T(A\mathbf{w}_1+\mathbf{e}_1b_1)+c_1\mathbf{e}_2^T\xi_2, \\ &\text{s.a.} \begin{cases} -(B\mathbf{w}_2+\mathbf{e}_2b_1)+\xi_2 \geq \mathbf{e}_2 \\ \xi_2 \geq 0 \end{cases}, \end{aligned} \tag{4-2}$$

e

$$\begin{aligned} &\min_{\mathbf{w}_2,b_2,\xi_1} \frac{1}{2}(B\mathbf{w}_2+\mathbf{e}_2b_2)^T(B\mathbf{w}_2+\mathbf{e}_2b_2)+c_2\mathbf{e}_1^T\xi_1, \\ &\text{s.a.} \begin{cases} (A\mathbf{w}_1+\mathbf{e}_1b_2)+\xi_1 \geq \mathbf{e}_1 \\ \xi_1 \geq 0 \end{cases}, \end{aligned} \tag{4-3}$$

em que $c_i > 0, i = 1,2$ são parâmetros de penalidade, $\mathbf{e}_i, i = 1,2$ são vetores de uns, com $\mathbf{e}_1 \in \mathbb{R}^p$, $\mathbf{e}_2 \in \mathbb{R}^q$ e $\xi_i, i = 1,2$ são vetores de folgas com $\xi_1 \in \mathbb{R}^p, \xi_2 \in \mathbb{R}^q$.

Semelhante ao realizado na Seção 3.2.1, escrevemos a Lagrangiana para o problema (4-2), omitiremos a resolução para o problema (4-3), porém é feito de modo análogo ao descrito a seguir.

$$\begin{aligned} L(\mathbf{w}_1,b_1,\xi_2,\alpha,\beta) =& \frac{1}{2}(A\mathbf{w}_1+\mathbf{e}_1b_1)^T(A\mathbf{w}_1+\mathbf{e}_1b_1)+c_1\mathbf{e}_2^T\xi_2+ \\ &+\alpha^T((B\mathbf{w}_1+\mathbf{e}_2b_1)-\xi_2+\mathbf{e}_2)+\beta^T(-\xi_2), \end{aligned} \tag{4-4}$$

em que $\alpha = (\alpha_1,\alpha_2,\ldots,\alpha_p)^T$ e $\beta = (\beta_1,\beta_2,\ldots,\beta_q)^T$ são vetores dos multiplicadores de Lagrange.

As condições $KKT$, que são necessárias e suficientes para otimalidade do problema (4-4), são apresentadas a seguir:

$$A^T(A\mathbf{w}_1+\mathbf{e}_1b_1)+B^T\alpha = 0, \tag{4-5}$$

$$e_1^T(A\mathbf{w}_1+\mathbf{e}_1b_1)+e_2^T\alpha = 0, \tag{4-6}$$

$$c_1\mathbf{e}_2-\alpha-\beta = 0, \tag{4-7}$$

$$-(B\mathbf{w}_1+\mathbf{e}_2b_1)+\xi_2 \geq \mathbf{e}_2, \quad \xi_2 \geq 0, \tag{4-8}$$

$$\alpha^T((B\mathbf{w}_1+\mathbf{e}_2b_1)-\xi_2+\mathbf{e}_2) = 0, \quad \beta^T(\xi_2) = 0, \tag{4-9}$$

$$\alpha \geq 0, \quad \beta \geq 0. \tag{4-10}$$

A fim de obter o problema dual de (4-2), foram utilizadas as Eqs. (4-5) e (4-6),

conforme é apresentado por Tomar e Agarwal [157], como segue:

$$\begin{bmatrix} A^T \\ e_1^T \end{bmatrix} \begin{bmatrix} A & e_1 \end{bmatrix} \begin{bmatrix} w_1 \\ b_1 \end{bmatrix} + \begin{bmatrix} B^T \\ e_2^T \end{bmatrix} \alpha = 0. \tag{4-11}$$

Para simplificar os cálculos, nomearemos $\begin{bmatrix} A & e_1 \end{bmatrix}$ por $G$, $\begin{bmatrix} B & e_2 \end{bmatrix}$ por $H$ e $\begin{bmatrix} \mathbf{w}_1 \\ b_1 \end{bmatrix}$ por $u_1$, deste modo, a Eq. (4-11) pode ser reformulado como segue:

$$(G^T G)u_1 + H^T \alpha = 0. \tag{4-12}$$

Isolando $u_1$ em (4-12), obtemos a Eq. (4-13):

$$u_1 = -(G^T G)^{-1} H^T \alpha. \tag{4-13}$$

Para obter o problema dual correspondente, o TWSVM assume que a matriz resultante de $(G^T G)^{-1}$ é não singular, caso isso não ocorra, é feito uma perturbação acrescentando $\varepsilon I$, em que $\varepsilon$ é um escalar positivo próximo de zero, a fim de manter a estrutura dos dados, $I$ é uma matriz identidade de dimensões apropriadas, desse modo $(G^T G)^{-1}$ podem ser trocadas por $(G^T G + \varepsilon I)^{-1}$ em (4-13), conforme sugere Shao et al. [146].

O Wolfe dual [98] do problema (4-2) é obtido fazendo as devidas substituições e algumas operações utilizando a Lagrangiana (4-4), juntamente com as condições $KKT$ (4-5↔4-10) conforme apresentado a seguir:

$$\begin{aligned} &\max_{\alpha} \quad \mathbf{e}_2^T \alpha - \frac{1}{2} \alpha^T H (G^T G + \varepsilon I)^{-1} H^T \alpha, \\ &\text{s.a.} \left\{ 0 \leq \alpha \leq c_1 \mathbf{e}_2. \right. \end{aligned} \tag{4-14}$$

Note que $G = [A \quad e_1] \in \mathbb{R}^{p \times (n+1)}$, assim $(G^T G + \varepsilon I)^{-1}$ tem as seguintes dimensões $(n+1) \times (n+1)$, indicando que o TWSVM trabalha bem quando o valor de $n$ é pequeno, porém se torna impraticável para problemas com alta dimensão.

Dada uma nova amostra $\mathbf{x} \in \mathbb{R}^n$, ela pertencerá à classe cuja menor distância seja obtida pela Eq. (4-15).

$$\text{Classe i} = \min_{i=1,2} |\mathbf{w}_i^T \mathbf{x} + b_i|, \tag{4-15}$$

em que $|\cdot|$ é a distância perpendicular da amostra $\mathbf{x}$ aos planos $\mathbf{w}_i^T \mathbf{x} + b_i = 0, \quad i = 1, 2$.

A Fig. (4.1) apresenta o comportamento dos hiperplanos utilizando o TWSVM para dados linearmente separáveis.

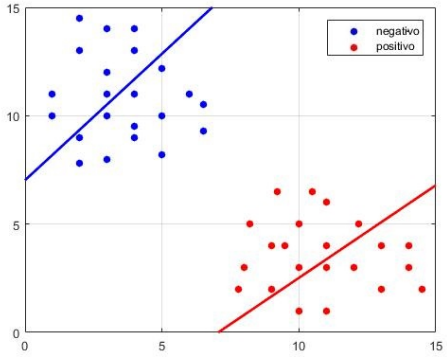



**Fig 4.1:** *Gráfico TWSVM linearmente separável.*

Um estudo comparativo entre o SVM, GEPSVM e TWSVM é realizado em Jayadeva et al. [66].

## 4.4 TWSVM não linear

O TWSVM também foi desenvolvido para trabalhar com dados não linearmente separáveis, isso é feito utilizando a função *kernel* que mapeia as amostras do espaço de origem para o espaço de alta dimensão, denominado espaço de características, conforme descrito na Seção 3.2.4. Ao invés de planos, são considerados superfícies geradas por *kernel*, conforme é apresentado a seguir:

$$K(x^T, D^T)u_1 + b_1 = 0 \quad \text{e} \quad K(x^T, D^T)u_2 + b_2 = 0, \tag{4-16}$$

em que $D = [A \quad B]^T$, $K$ é um *kernel* escolhido apropriadamente.

De modo análogo ao realizado na Seção 4.3, desenvolveremos os cálculos para um dos casos do problema primal para o TWSVM não linear, conforme segue:

$$\begin{aligned} &\min_{u_1,b_1,\xi} \frac{1}{2}(K(A,D^T)u_1 + e_1b_1)^T(K(A,D^T)u_1 + e_1b_1) + c_1e_2^T\xi, \\ &\text{s.a.} \left\{ -(K(B,D^T)u_1 + e_2b_1) + \xi \geq e_2, \quad \xi \geq 0, \right. \end{aligned} \tag{4-17}$$

com o parâmetro $c_1 > 0$. A Lagrangiana é apresentada a seguir:

$$L(u_1,b_1,\xi,\alpha,\beta) = \frac{1}{2}(K(A,D^T)u_1 + e_1b_1)^T(K(A,D^T)u_1 + e_1b_1) + \\ + c_1e_2^T\xi + \alpha^T((K(B,D^T)u_1 + e_2b_1) - \xi + e_2) + \\ + \beta^T(-\xi). \tag{4-18}$$

As condições $KKT$ de (4-18) são:

$$K(A,D^T)^T(K(A,D^T)\mathbf{u}_1 + \mathbf{e}_1b_1) + K(B,D^T)^T\alpha = 0, \tag{4-19}$$
$$e_1^T(K(A,D^T)\mathbf{u}_1 + \mathbf{e}_1b_1) + e_2^T\alpha = 0, \tag{4-20}$$
$$c_1\mathbf{e}_2 - \alpha - \beta = 0, \tag{4-21}$$
$$-(K(B,D^T)\mathbf{u}_1 + \mathbf{e}_2b_1) + \xi_2 \geq \mathbf{e}_2, \quad \xi_2 \geq 0, \tag{4-22}$$
$$\alpha^T((K(B,D^T)\mathbf{u}_1 + \mathbf{e}_2b_1) - \xi_2 + \mathbf{e}_2) = 0, \quad \beta^T(\xi_2) = 0, \tag{4-23}$$
$$\alpha \geq 0, \quad \beta \geq 0. \tag{4-24}$$

Com o objetivo de obter o problema dual de (4-17), foi realizado uma fusão das Eqs. (4-19) e (4-20) conforme é apresentado a seguir:

$$\begin{bmatrix} K(A,D^T)^T \\ e_1^T \end{bmatrix} \begin{bmatrix} K(A,D^T) & e_1 \end{bmatrix} \begin{bmatrix} u_1 \\ b_1 \end{bmatrix} + \begin{bmatrix} K(B,D^T)^T \\ e_2^T \end{bmatrix} \alpha = 0. \tag{4-25}$$

Semelhantemente ao realizado na Seção 4.3, com o intuito de simplificar os cálculos, denominaremos $\begin{bmatrix} K(A,D^T) & e_1 \end{bmatrix}$ por $P$, $\begin{bmatrix} K(B,D^T) & e_2 \end{bmatrix}$ por $Q$ e $\begin{bmatrix} \mathbf{u}_1 \\ b_1 \end{bmatrix}$ por $z_1$, desse modo, a Eq. (4-25) pode ser reformulada como segue:

$$(P^TP)z_1 + Q^T\alpha = 0. \tag{4-26}$$

Isolando $z_1$ em (4-26), obtemos a Eq. (4-27) como segue:

$$z_1 = -(P^TP)^{-1}Q^T\alpha. \tag{4-27}$$

Uma nova amostra de dados é classificada utilizando (4-28),

$$\text{Classe i} = \min_{i=1,2} |K(x^T,D^T)u_i + b_i|. \tag{4-28}$$

Em Jayadeva et al. [66], foi apresentado um comparativo no desempenho do tempo computacional, bem como a acurácia em bases de dados do repositório da UCI

[87].

## 4.5 Extensão baseadas no TWSVM

Com o intuito de melhorar o desempenho da abordagem original do TWSVM, pesquisadores têm proposto algumas extensões baseadas no TWSVM. Dentre essas extensões, uma delas será apresentada a seguir, pois essa extensão implementa o princípio de minimização do risco estrutural, por acrescentar o termo de regularização em seu problema primal.

### 4.5.1 TBSVM

Em 2011, Shao et al. [146] propuseram uma alteração no TWSVM, que foi denominada *Twin Bounded Support Vector Machine* (TBSVM), onde foi acrescentado o termo de regularização (4-29), de modo a minimizar os erros no conjunto de treinamento,

$$\frac{1}{2}c_3(\| \mathbf{w}_1 \|^2 + b_1^2). \tag{4-29}$$

Tomar e Agarwal [157], em um estudo comparativo entre o TWSVM e algumas de suas extensões, afirmam que o TBSVM é mais efetivo em termos de tempo computacional e acurácia, no processo de classificação.

### 4.5.2 Caso Linear do TBSVM

Os problemas primais para o caso linear do TBSVM é apresentado a seguir:

$$\begin{aligned} &\min_{\mathbf{w}_1,b_1,\xi_2} \frac{1}{2}c_3(\| w_1 \|^2 + b_1^2) + \frac{1}{2}(A\mathbf{w}_1 + \mathbf{e}_1 b_1)^T (A\mathbf{w}_1 + \mathbf{e}_1 b_1) + c_1 \mathbf{e}_2^T \xi_2, \\ &\text{s.a.} \begin{cases} -(B\mathbf{w}_2 + \mathbf{e}_2 b_1) + \xi_2 \geq \mathbf{e}_2 \\ \xi_2 \geq 0 \end{cases}, \end{aligned} \tag{4-30}$$

e

$$\begin{aligned} &\min_{\mathbf{w}_2,b_2,\xi_1} \frac{1}{2}c_4(\| w_2 \|^2 + b_2^2) + \frac{1}{2}(B\mathbf{w}_2 + \mathbf{e}_2 b_2)^T (B\mathbf{w}_2 + \mathbf{e}_2 b_2) + c_2 \mathbf{e}_1^T \xi_1, \\ &\text{s.a.} \begin{cases} (A\mathbf{w}_1 + \mathbf{e}_1 b_2) + \xi_1 \geq \mathbf{e}_1 \\ \xi_1 \geq 0 \end{cases}, \end{aligned} \tag{4-31}$$

em que $c_i > 0, i = 1, \ldots, 4$ são parâmetros de penalidade, $\frac{1}{2}c_3(\| \mathbf{w}_1 \|^2 + b_1^2)$ e $\frac{1}{2}c_4(\| \mathbf{w}_2 \|^2 + b_2^2)$ são termos de regularização que minimizam os erros de treinamento.

Semelhante ao realizado na Seção 4.3 para o TWSVM, escrevemos a Lagrangiana para o problema (4-30), omitiremos a resolução para o problema (4-31), porém é feito de modo análogo ao descrito a seguir:

$$L(\mathbf{w}_1, b_1, \xi_2, \alpha, \beta) = \frac{1}{2} c_3 (\| w_1 \|^2 + b_1^2) + \frac{1}{2} (A\mathbf{w}_1 + \mathbf{e}_1 b_1)^T (A\mathbf{w}_1 + \mathbf{e}_1 b_1) + c_1 \mathbf{e}_2^T \xi_2 + \\ + \alpha^T ((B\mathbf{w}_1 + \mathbf{e}_2 b_1) - \xi_2 + \mathbf{e}_2) + \beta^T (-\xi_2), \tag{4-32}$$

em que $\alpha = (\alpha_1, \alpha_2, \ldots, \alpha_p)^T$ e $\beta = (\beta_1, \beta_2, \ldots, \beta_q)^T$ são vetores dos multiplicadores de Lagrange.

As condições $KKT$, que são necessárias e suficientes para otimalidade do problema (4-32) são apresentadas a seguir:

$$c_3 w_1 + A^T (A\mathbf{w}_1 + \mathbf{e}_1 b_1) + B^T \alpha = 0, \tag{4-33}$$
$$c_3 b_1 + e_1^T (A\mathbf{w}_1 + \mathbf{e}_1 b_1) + e_2^T \alpha = 0, \tag{4-34}$$
$$c_1 \mathbf{e}_2 - \alpha - \beta = 0, \tag{4-35}$$
$$-(B\mathbf{w}_1 + \mathbf{e}_2 b_1) + \xi_2 \geq \mathbf{e}_2, \quad \xi_2 \geq 0, \tag{4-36}$$
$$\alpha^T ((B\mathbf{w}_1 + \mathbf{e}_2 b_1) - \xi_2 + \mathbf{e}_2) = 0, \quad \beta^T (\xi_2) = 0, \tag{4-37}$$
$$\alpha \geq 0, \quad \beta \geq 0. \tag{4-38}$$

Trabalhando com as Eqs. (4-33) e (4-34), temos:

$$c_3 \begin{bmatrix} w_1 \\ b_1 \end{bmatrix} + \begin{bmatrix} A^T \\ e_1^T \end{bmatrix} \begin{bmatrix} A & e_1 \end{bmatrix} \begin{bmatrix} w_1 \\ b_1 \end{bmatrix} + \begin{bmatrix} B^T \\ e_2^T \end{bmatrix} \alpha = 0. \tag{4-39}$$

Para simplificar os cálculos, nomearemos: $v_1 = \begin{bmatrix} \mathbf{w}_1 \\ b_1 \end{bmatrix}$, $G = \begin{bmatrix} A & e_1 \end{bmatrix}$ e $H = \begin{bmatrix} B^T \\ e_2^T \end{bmatrix}$. Deste modo, a Eq. (4-39) pode ser reformulada como segue:

$$v_1 = -(G^T G + c_3 I)^{-1} H^T \alpha. \tag{4-40}$$

Do mesmo modo obtemos:

$$v_2 = -(H^T H + c_4 I)^{-1} G^T \gamma, \tag{4-41}$$

em que $\gamma$ é um multiplicador de Lagrange.

O problema dual correspondente aos problemas (4-30) e (4-31), respectivamente, é dado por (4-42) e (4-43):

$$\begin{aligned} \max_{\alpha} \quad & \mathbf{e}_2^T\alpha - \frac{1}{2}\alpha^T H(G^T G + c_3 I)^{-1} H^T \alpha, \\ \text{s.a.} \quad & \{0 \leq \alpha \leq c_1 \end{aligned} \tag{4-42}$$

e

$$\begin{aligned} \max_{\gamma} \quad & \mathbf{e}_1^T\gamma - \frac{1}{2}\gamma^T G(H^T H + c_4 I)^{-1} G^T \gamma, \\ \text{s.a.} \quad & \{0 \leq \gamma \leq c_2 \quad . \end{aligned} \tag{4-43}$$

Segundo Shao et al. [146], embora a formulação seja a mesma para o TWSVM Eq. (4-14), quando substituído o parâmetro $c_3$ em (4-42) por $\varepsilon$, existe diferença em seu significado. Enquanto o parâmetro $\varepsilon$ é um escalar fixo bem pequeno, o parâmetro $c_3$ é um fator de ponderação que determina um *tradeoff* entre o termo de regularização e o risco empírico.

Para a classificação de uma nova amostra é utilizada (4-44) como segue:

$$\text{Classe i} = arg \min_{(j=1,2)} \frac{| w_j^T x + b_j |}{\| w_j \|}, \tag{4-44}$$

a menor distância determina em qual classe a nova amostra pertencerá.

Os problemas de otimização do TBSVM (4-30) e (4-31) são resolvidos usando a técnica *Successive Over Relaxation (SOR)* [99] com o objetivo de melhorar a velocidade dos dados em treinamento na resolução de problemas de vetores de suporte. O algoritmo SOR é um procedimento iterativo que emprega as iterações de Gauss-Seidel (GS) com o fator de extrapolação $t \in (0,2)$ para acelerar a solução dos sistemas lineares e PPQs com convergência linear [139].

Podemos rescrever (4-42) ou (4-43) como segue:

$$\begin{aligned} \max_{\alpha} \quad & e^T\alpha - \frac{1}{2}\alpha^T Q\alpha, \\ \text{s.a.} \quad & 0 \leq \alpha \leq c, \end{aligned} \tag{4-45}$$

em que $Q \in \mathbb{R}^{m\times m}$

O problema (4-45) pode ser resolvido eficientemente, como realizado em [99, 146, 158, 139, 112], utilizando o Algoritmo (4.5.1) apresentado a seguir:

**Algoritmo 4.5.1** *Algoritmo SOR [99]*

1. *Escolha* $p \in (0,2)$*;*
2. *Inicialize* $\omega = [\omega_1, \ldots, \omega_n]'$*;*
3. *Obtenha* $L \in \mathbb{R}^{m \times m}$*, onde os elementos diferentes de zero de L constituem a parte da matriz triangular estritamente inferior da matriz simétrica Q;*
4. *Obtenha* $E \in \mathbb{R}^{m \times m}$*, onde os elementos diferentes de zero de E constituem a diagonal da matriz Q;*
5. *Calcule* $\omega^{i+1}$ *como segue:*
   $\omega_{i+1} = (\omega^i - pE^{-1}(Q\omega^i - e + L(\omega^{i+1} - \omega^i)))$*;*
6. *Continue até* $\| \omega^{i+1} - \omega^i \|$ *seja menor que um determinado valor de tolerância.*

---

## 4.6 Considerações Finais

Nos últimos anos, muitas variações do TWSVM foram propostas, tais como: *Fuzzy* TWSVM [70], *Least squares* TWSVM [79], ν-TWSVM [116, 178], TWSVM *regression* [117], *robust* TWSVM [127], ITSVM [152], NHSVM [139], *Linear Programming ITSVM* (LPTWSVM) [85], entre outros. Mais detalhes sobre extensões do TWSVM podem ser encontradas em Jayadeva et al. [67]. Devido ao extenso número de algoritmos propostos, não detalhamos como trabalha cada algoritmo, nem a construção matemática por trás de cada um deles, seguimos uma lógica conforme o delineamento do nosso trabalho, por isso apresentamos somente o TBSVM.

CAPÍTULO 5

# Máquinas de vetores de suportes duplo para problemas de múltiplas classes

## 5.1 Considerações Iniciais

Neste capítulo, apresentamos a construção de extensões do TWSVM para problemas de classificação de múltiplas classes. Embora o TWSVM para problemas de múltiplas classes tenha apresentado grandes progressos, ainda há poucos estudos sobre as comparações dos diferentes classificadores de múltiplas classes baseado nesse algoritmo [39]. Outro problema é como escolher o melhor método, ou o mais adequado para um problema prático de classificação de múltiplas classes. Os problemas de múltiplas classes, utilizando o SVM, podem ser divididos em duas categorias. A primeira contempla o método "*all together*", em que os problemas de múltiplas classes são realizados resolvendo um grande problema de programação quadrática (PPQ), o que acaba atraindo pouca atenção, devido a sua complexidade assintótica. A outra categoria contempla o método de decomposição-reconstrução, que resolve o problema de classificação de múltiplas classes em uma série de problemas de classificação binária. No processo prático de classificação de múltiplas classes, o método de decomposição-reconstrução é o mais utilizado [61, 39].

## 5.2 Estratégias para problemas de múltiplas classes

As estratégias que podem ser usadas para estender o TWSVM binário para TWSVM de múltiplas classes incluem as seguintes estratégias: "$1-\times-1$", "$1-\times-$ *todos*", "$1-\times-1-\times-$ *restante*", "*todos* $-\times-1$", método baseado em *"árvores binárias"* e também o método *"gráfico acíclico dirigido"* desenvolvido a partir da estratégia "$1-\times-1-\times-$ *restante*".

Nesta seção, iremos descrever algumas das estratégias para problemas de múltiplas classes, que estão alinhadas com o foco do nosso trabalho. Mais detalhes sobre as estratégias podem ser encontradas em Ding et al. [39] e Jayadeva et al. [67].

A primeira estratégia que iremos explanar é a "$1-\times-$ *restante*". Esse foi o primeiro método usado para estender o TWSVM para resolver problemas de classificação com várias classes, conforme apresentado em Xie et al. [172]. Essa abordagem leva facilmente ao problema do desequilíbrio de classes e produz um desempenho ruim.

Para um problema de classificação de $K$ classes, a abordagem gera $K$ hiperplanos, um hiperplano para cada classe. Cada hiperplano corresponde a um PPQ. Deste modo, a estratégia "$1-\times-$ *restante*" para problemas de múltiplas classes do TWSVM precisa resolver $K$ PPQs, o que também gera um custo computacional muito elevado, quando se trata de grandes bases de dados.

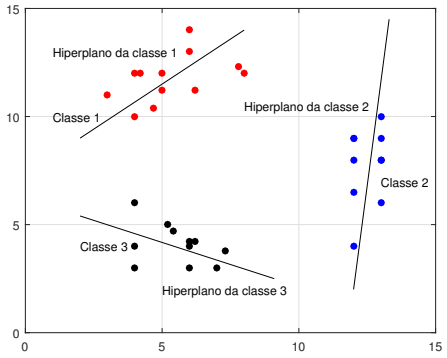



**Fig 5.1:** *Ilustração da estratégia* "$1-\times-$ *restante*".

Para a estratégia denominada "$1-\times-1$", proposta inicialmente por Hsu e Lin [61], para problemas de classificação com múltiplas classes utilizando o SVM, em um problema com $K$ classes, a abordagem gera um classificador binário para cada par de classes. Deste modo, esse método constrói $K(K-1)/2$ classificadores binários para o TWSVM. Cada classificador binário usa apenas amostras de duas classes e pode classificar as duas classes corretamente, porém como apenas dois tipos de amostras são envolvidos para cada classificador, e nenhuma informação é dada para as amostras restantes, então os resultados obtidos são desfavoráveis. Nessa abordagem é adotado o método *max-wins voting*, ou seja, a classe que obteve mais votos ganha.

Em 2003, Angulo et al. [4] propuseram um novo algoritmo de classificação para problemas de múltiplas classes, denominado *support vector classication-regression for K-class* (K-SVCR). Esse algoritmo de aprendizagem com saídas ternárias $\{-1,0,+1\}$ é baseado na teoria de vetores de suporte de Vapnik, e avalia todas as amostras de treinamento com uma estrutura "$1-\times-1-\times-$ *restante*" durante a fase de decomposição, fazendo uma combinação entre a formulação do SVM para classificação e regressão. Comparado com outros algoritmos de classificação para múltiplas classes, o K-SVCR obtém maior desempenho de generalização, ou seja, a maior capacidade de responder corretamente a

dados que não foram utilizados no processo de treinamento, pois todas as amostras são utilizadas na construção do hiperplano de classificação.

Já em 2013, Xu et al. [177] estenderam o TWSVM para problemas de múltiplas classes, denominado Twin-KSVC, utilizando algumas ideias das técnicas de correção de erros de códigos de saída (ECOC). O Twin-KSVC avalia todos os pontos de treinamento em uma estrutura "$1-\times-1-\times-$ *restante*". Nessa abordagem, dois tipos de amostras das $K$ classes são utilizadas na construção dos dois hiperplanos não paralelos. As amostras restantes são mapeadas em uma região entre os dois hiperplanos não paralelos. Como todas as amostras são utilizadas, o Twin-KSVC obtém maior poder de classificação quando comparado com o TWSVM utilizando o método "$1-\times-1$".

A seguir iremos descrever sobre o K-SVC, Twin-KSVC e LSTKSVC.

## 5.3 K-SVCR

K-SVCR [4], é um algoritmo com saídas ternárias $\{-1,0,1\}$ baseado na teoria de vetores de suporte de Vapnik [4], e introduz os vetores de suporte para classificação e regressão em problemas com $K$-classes. Este algoritmo avalia todos os dados de treinamento na fase de decomposição usando uma formulação de máquina de classificação e regressão em uma estrutura "$1-\times-1-\times-$ *restante*", conforme problema apresentado a seguir:

$$\min_{\mathbf{w},b,\xi,\gamma,\hat{\gamma}} \frac{1}{2} \parallel \mathbf{w} \parallel^2 + c_1 \sum_{i=1}^{m} \xi_i + c_2 \sum_{i=1}^{n} (\gamma_i + \hat{\gamma}_i),$$

$$\text{s.a.} \begin{cases} y_i(\mathbf{w}^T \phi(\mathbf{x}_i) + b) \geq 1 - \xi_i, \quad i = 1,2,\ldots,m \\ -\delta - \hat{\gamma} < \mathbf{w}^T \phi(\mathbf{x}_i) + b, \quad i = 1,2,\ldots,n \\ \mathbf{w}^T \phi(\mathbf{x}_i) + b \leq \delta + \gamma, \quad i = 1,2,\ldots,n \\ \xi_i, \gamma_i, \hat{\gamma}_i \geq 0 \end{cases},$$

em que $\xi_i, \gamma_i$ e $\hat{\gamma}_i$ são variáveis de folgas, $m$ é o número de padrões pertencentes às duas classes de rótulos que precisam ser separadas, e $n$ é o número de padrões rotulados por "0" pertencentes as outras classes. O parâmetro positivo $\delta$ é restrito e deve ser menor do que 1, para evitar sobreposição.

A função de decisão do hiperplano pode ser reescrita como apresentado a seguir:

$$f(\mathbf{x}) = \begin{cases} +1 & \text{se} \quad \sum_{i=1}^{N_{SV}} \alpha_i K(\mathbf{x}_i, \mathbf{x}) + b \geq \delta \\ -1 & \text{se} \quad \sum_{i=1}^{N_{SV}} \alpha_i K(\mathbf{x}_i, \mathbf{x}) + b \leq -\delta \\ 0 & \text{caso contrário} \end{cases},$$

em que $\alpha_i$ são multiplicadores de Lagrange, $N_{SV}$ é o número de vetores de suporte. Diferentemente do TWSVM, o K-SVCR é semelhante ao SVM porque precisa resolver um único PPQ de tamanho maior. Além disso, os dois tipos de amostras são usados uma vez nas restrições, enquanto as amostras restantes são usadas duas vezes nas restrições, o que leva a uma baixa velocidade computacional.

### 5.3.1 Twin-KSVC

O algoritmo Twin-KSVC [177], semelhante ao K-SVCR, avalia as amostras de treinamento em uma estrutura "$1-\times-1-\times-restante$", com saídas ternárias $\{-1,0,1\}$, sendo dois tipos de amostras selecionadas das $K$ classes, com o objetivo de encontrar dois planos não paralelos. As amostras restantes são mapeadas para uma região entre os dois planos não paralelos construídos. A seguir iremos descrever o caso linear. Para mais detalhes e também sobre a extensão para o caso não linear, podem ser encontradas em Xu et al. [177].

#### Caso linear Twin-KSVC

Denotaremos por $A \in \mathbb{R}^{m_1 \times n}$ e $B \in \mathbb{R}^{m_2 \times n}$ as amostras pertencentes às classes rotuladas por "$+1$" e "$-1$", respectivamente. Denotaremos por $C \in \mathbb{R}^{m_3 \times n}$ as amostras que serão rotuladas por "0", e serão consideradas as amostras restantes que serão mapeadas para a região entre os hiperplanos, satisfazendo as seguintes restrições: $-(C\mathbf{w}_1 + \mathbf{e}_3 b_1) \geq \mathbf{e}_3(1-\varepsilon)$ e $C\mathbf{w}_2 + \mathbf{e}_3 b_2 \geq \mathbf{e}_3(1-\varepsilon)$, em que $\varepsilon$ é um parâmetro positivo escolhido previamente.

Os dois hiperplanos não paralelos são definidos da seguinte maneira:

$$\mathbf{w}_1^T \mathbf{x} + b_1 = 0 \quad \text{e} \quad \mathbf{w}_2^T \mathbf{x} + b_2 = 0, \tag{5-1}$$

e são obtidos ao resolver os problemas de programação quadrática, conforme apresentado

a seguir:

$$\min_{\mathbf{w}_1,b_1,\xi,\eta} \frac{1}{2} \parallel A\mathbf{w}_1 + \mathbf{e}_1 b_1 \parallel^2 + c_1 \mathbf{e}_2^T \xi + c_2 \mathbf{e}_3^T \eta,$$
$$\text{s.a.} \begin{cases} -\left(B\mathbf{w}_1 + \mathbf{e}_2 b_1\right) + \xi \geq \mathbf{e}_2 \\ -\left(C\mathbf{w}_1 + \mathbf{e}_3 b_1\right) + \eta \geq \mathbf{e}_3(1-\varepsilon) \\ \eta \geq 0\mathbf{e}, \quad \xi \geq 0\mathbf{e} \end{cases}, \tag{5-2}$$

e

$$\min_{\mathbf{w}_2,b_2,\xi^*,\eta^*} \frac{1}{2} \parallel B\mathbf{w}_2 + \mathbf{e}_2 b_2 \parallel^2 + c_3 \mathbf{e}_1^T \xi^* + c_4 \mathbf{e}_3^T \eta^*,$$
$$\text{s.a.} \begin{cases} -\left(A\mathbf{w}_2 + \mathbf{e}_1 b_2\right) + \xi^* \geq \mathbf{e}_1 \\ -\left(C\mathbf{w}_2 + \mathbf{e}_3 b_2\right) + \eta^* \geq \mathbf{e}_3(1-\varepsilon) \\ \eta^* \geq 0\mathbf{e}, \quad \xi^* \geq 0\mathbf{e} \end{cases}, \tag{5-3}$$

em que $\mathbf{w}_1$ e $\mathbf{w}_2 \in \mathbb{R}^{n\times 1}$, $b_1$ e $b_2 \in \mathbb{R}^{1\times 1}$, $\xi \in \mathbb{R}^{m_2\times 1}$, $\eta \in \mathbb{R}^{m_3\times 1}$, $\xi^* \in \mathbb{R}^{m_1\times 1}$, $\eta^* \in \mathbb{R}^{m_3\times 1}$, $\mathbf{e}_1 \in \mathbb{R}^{m_1\times 1}$, $\mathbf{e}_2 \in \mathbb{R}^{m_2\times 1}$ e $\mathbf{e}_3 \in \mathbb{R}^{m_3\times 1}$.

Sem perda de generalidade, iremos descrever a Lagrangiana somente relativa ao problema (5-2).

$$\begin{aligned} L(\mathbf{w}_1,b_1,\eta,\xi) = &\frac{1}{2} \parallel A\mathbf{w}_1 + \mathbf{e}_1 b_1 \parallel^2 + c_1 \mathbf{e}_2^T \xi + c_2 \mathbf{e}_3^T \eta - \alpha^T \Bigg( -\left(B\mathbf{w}_1 + \mathbf{e}_2 b_1\right) + \\ &+ \xi - \mathbf{e}_2 \Bigg) - \beta^T \Bigg( -\left(C\mathbf{w}_1 + \mathbf{e}_3 b_1\right) + \eta - \mathbf{e}_3 + \varepsilon_3) \Bigg) - \lambda^T \eta - \\ &- \sigma^T \xi, \end{aligned} \tag{5-4}$$

em que $\alpha, \beta, \lambda, \sigma \geq 0\mathbf{e}$, são multiplicadores de Lagrange.

Resolvendo o problema (5-2) obtemos:

$$u = [\mathbf{w}_1; b_1] = (H^T H + \delta I)^{-1}(G^T \alpha + M^T \beta), \tag{5-5}$$

em que $H = [A \quad \mathbf{e}_1]$, $G = [B \quad \mathbf{e}_2]$, $M = [C \quad \mathbf{e}_3]$, $\delta$ é um valor positivo bem pequeno para não influenciar tanto na estrutura da matriz, $I$ é uma matriz identidade de dimensões apropriadas, o termo $\delta I$ na Eq. (5-5) é para garantir que a matriz $(H^T H + \delta I)^{-1}$ seja não singular. Ao realizarmos algumas substituições dos valores obtidos derivando a

Lagrangiana e, fazendo as devidas substituições em (5-4), obtemos:

$$L = \frac{1}{2}\gamma^T N(H^T H)^{-1} N^T \gamma + \mathbf{e}_4^T \gamma, \tag{5-6}$$

em que $\mathbf{e}_4 = [\mathbf{e}_2; \mathbf{e}_3(1-\varepsilon)]$, $N = [G; M]$, $\gamma = [\alpha; \beta]$.

A formulação dual para o problema (5-2) é apresentado a seguir:

$$\begin{aligned} &\max_{\gamma} \frac{1}{2}\gamma^T N(H^T H)^{-1} N^T \gamma + \mathbf{e}_4^T, \\ &\text{s.a.} \left\{ 0 \leq \gamma \leq F, \right. \end{aligned} \tag{5-7}$$

em que $F = [c_1\mathbf{e}_2; c_2\mathbf{e}_3]$.

De modo análogo ao realizado para o problema (5-2), obtemos $u^*$ para o problema (5-3), conforme apresentado a seguir:

$$u^* = [\mathbf{w}_2; b_2] = (G^T G + \delta I)^{-1} P^T \rho, \tag{5-8}$$

em que $P = [H; M]$.

Para um novo ponto de teste $\mathbf{x}_i$, o Twin-KSVC determina seu rótulo de classe pela seguinte função de decisão,

$$f(\mathbf{x}_i) = \begin{cases} +1, & se \quad \mathbf{w}_1^T \mathbf{x}_i + \mathbf{e}b_1 > -1 + \varepsilon \\ -1, & se \quad \mathbf{w}_2^T \mathbf{x}_i + \mathbf{e}b_2 < 1 - \varepsilon \\ 0, & \text{caso contrário} \end{cases} .$$

Uma das vantagens do Twin-KSVC é que consome quase o mesmo tempo de execução que a estrutura "$1 \times restante$" do TWSVM, além disso, requer menos tempo computacional do que o algoritmo K-SVCR.

## 5.4 LSTKSVC

A versão mínimos quadrados para problemas de múltiplas classes TWSVM, que será denotado por LSTKSVC [109], é um algoritmo recente para resolução de problemas de múltiplas classes baseado no PSVM [43] e Twin-KSVC [177]. Como no K-SVCR e no Twin-KSVC, o algoritmo proposto avalia todos os pontos de treinamento em uma estrutura "$1 - \times - 1 - \times - restante$" com saída ternária $\{-1, 0, +1\}$.

Seja a matriz $A \in \mathbb{R}^{m_1 \times n}$ que contempla as amostras pertencentes à classe rotuladas por "$+1$", $B \in \mathbb{R}^{m_2 \times n}$ contemplando as amostras pertencentes à classe de

rótulos " $-1$" e $C \in \mathbb{R}^{m_3 \times n}$ contemplando as amostras restantes rotulados por "0". Os dois hiperplanos não paralelos são definidos da seguinte maneira:

$$\mathbf{w}_1^T \mathbf{x} + b_1 = 0 \quad \text{e} \quad \mathbf{w}_2^T \mathbf{x} + b_2 = 0. \tag{5-9}$$

A função de decisão do LSTKSVC é obtida diretamente pelo problema primal conforme apresentado a seguir:

$$\begin{aligned} &\min_{\mathbf{w}_1, b_1} \frac{1}{2} \| A\mathbf{w}_1 + \mathbf{e}_1 b_1 \|^2 + \frac{c_1}{2} \lambda^T \lambda + \frac{c_2}{2} \theta^T \theta, \\ &\text{s.a.} \begin{cases} -\left(B\mathbf{w}_1 + \mathbf{e}_2 b_1\right) + \lambda = \mathbf{e}_2 \\ -\left(C\mathbf{w}_1 + \mathbf{e}_3 b_1\right) + \theta = \mathbf{e}_3 (1 - \varepsilon) \end{cases}, \end{aligned} \tag{5-10}$$

e

$$\begin{aligned} &\min_{\mathbf{w}_2, b_2} \frac{1}{2} \| B\mathbf{w}_2 + \mathbf{e}_2 b_2 \|^2 + \frac{c_3}{2} \lambda^T \lambda + \frac{c_4}{2} \theta^T \theta, \\ &\text{s.a.} \begin{cases} \left(A\mathbf{w}_2 + \mathbf{e}_1 b_2\right) + \lambda = \mathbf{e}_1 \\ \left(C\mathbf{w}_2 + \mathbf{e}_3 b_2\right) + \theta = \mathbf{e}_3 (1 - \varepsilon) \end{cases}, \end{aligned} \tag{5-11}$$

em que $\varepsilon$ é uma parâmetro positivo escolhido previamente, $\mathbf{e}_i$ com $i = 1, 2,$ e 3 são vetores cujos elementos são todos iguais a um e com dimensão apropriada.

Calculando o gradiente dos problemas (5-10) e (5-11) em relação a $\mathbf{w}_1$, $b_1$, $\mathbf{w}_2$ e $b_2$ e igualando a zero e organizando os resultados, obtemos: (5-12) e (5-13), conforme apresentado a seguir:

$$u_1 = -\left(E^T E + c_1 F^T F + c_2 G^T G\right)^{-1} \left(c_1 F^T \mathbf{e}_2 + c_2 G^T \mathbf{e}_3 (1 - \varepsilon)\right) \tag{5-12}$$

e

$$u_2 = -\left(F^T F + c_3 E^T E + c_4 G^T G\right)^{-1} \left(c_3 E^T \mathbf{e}_1 + c_4 G^T \mathbf{e}_3 (1 - \varepsilon)\right), \tag{5-13}$$

em que $E = \begin{bmatrix} A & \mathbf{e}_1 \end{bmatrix}$, $F = \begin{bmatrix} B & \mathbf{e}_2 \end{bmatrix}$, $G = \begin{bmatrix} C & \mathbf{e}_3 \end{bmatrix}$, $u_1 = \begin{bmatrix} \mathbf{w}_1 \\ b_1 \end{bmatrix}$ e $u_2 = \begin{bmatrix} \mathbf{w}_2 \\ b_2 \end{bmatrix}$.

Os hiperplanos não paralelos (5-9) serão obtidos das soluções dos PPQs (5-12) e (5-13), respectivamente. O rótulo final para uma amostra de teste $\mathbf{x}_i$ é decidido pela regra do voto e avaliado pela função de decisão para o caso linear como segue:

$$f(\mathbf{w}_i) = \begin{cases} +1, & se \quad \mathbf{w}_1^T \mathbf{x}_i + \mathbf{e}b_1 > -1 + \varepsilon \\ -1, & se \quad \mathbf{w}_2^T \mathbf{x}_i + \mathbf{e}b_2 < 1 - \varepsilon \\ 0, & \text{caso contrário} \end{cases} .$$

## 5.5 Considerações Finais

Nesse capítulo, foram apresentados alguns dos principais trabalhos que nos motivaram aos estudos das extensões do TWSVM.

CAPÍTULO 6

# Melhorias na versão mínimos quadrados para problemas de múltiplas classes do TWSVM

## 6.1 Considerações Iniciais

Neste capítulo, desenvolvemos um novo algoritmo de classificação para problemas de múltiplas classes, que utiliza uma estrutura "$1-\times-1-\times-$ *restante*", gerando saídas ternárias $\{-1,0,+1\}$. O problema de otimização a ser resolvido se resume em resolver um sistema de equações lineares ao invés de um problema de programação quadrática, como ocorre em algumas das extensões do TWSVM. Este estudo resultou em um artigo publicado no *Journal Neurocomputing* com o título: *Improvements on least squares twin multi-class classification support vector machine* [90].

## 6.2 Melhorias na versão mínimos quadrados para problemas de múltiplas classes envolvendo twin support vector machine

Motivado pelos estudos desenvolvidos por Nasiri et al. [109] que propôs o LSTKSVC, neste capítulo é apresentado um algoritmo de classificação para problemas de múltiplas classes [90].

Considerando $A \in \mathbb{R}^{m_1 \times n}$ representando as amostras de treinamento pertencentes à classe "$+1$", $B \in \mathbb{R}^{m_2 \times n}$ representando as amostras pertencentes à classe "$-1$" e $C \in \mathbb{R}^{m_3 \times n}$ pertencentes às amostras restantes, que pertencerão à classe rotulado por "0". O algoritmo avalia os pontos de treinamento em uma estrutura "$1-\times-1-\times-$ *restante*". A Fig. (6.1) ilustra o comportamento do algoritmo proposto para problemas de múltiplas classes.

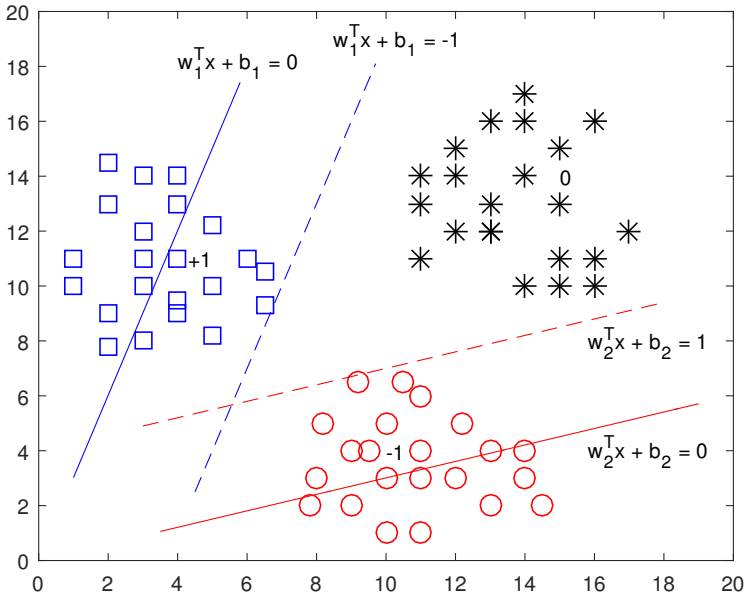



**Fig 6.1:** *Interpretação geométrica do comportamento do algoritmo.*

A seguir iremos descrever a matemática envolvida em relação aos casos lineares e não lineares de cada algoritmo.

### 6.2.1 Caso linear

Análogo ao feito para o LSTKSVC, o algoritmo proposto gera dois hiperplanos não paralelos para os dois tipos de amostras selecionadas em $K$ classes. Os restantes das amostras são mapeadas para a região entre os dois hiperplanos não paralelos, que são definidos na Eq. (6-1) a seguir:

$$\mathbf{w}_1^T\mathbf{x}+b_1=0 \quad \text{e} \quad \mathbf{w}_2^T\mathbf{x}+b_2=0, \tag{6-1}$$

em que os valores de $\mathbf{w}_1, \mathbf{w}_2, b_1$ e $b_2$ são obtidos por meio da solução do seguinte par de problemas de programação quadrática:

$$\begin{aligned} &\min_{\mathbf{w}_1,b_1} \frac{1}{2}c_5(\| \mathbf{w}_1 \|^2 + b_1^2) + \frac{1}{2} \| A\mathbf{w}_1 + \mathbf{e}_1 b_1 \|^2 + \frac{c_1}{2}\lambda^T\lambda + \frac{c_2}{2}\theta^T\theta, \\ &\text{s.a.} \begin{cases} -\left(B\mathbf{w}_1 + \mathbf{e}_2 b_1\right) + \lambda = \mathbf{e}_2 \\ -\left(C\mathbf{w}_1 + \mathbf{e}_3 b_1\right) + \theta = \mathbf{e}_3(1-\varepsilon) \end{cases}, \end{aligned} \tag{6-2}$$

e

$$\min_{\mathbf{w}_2,b_2} \frac{1}{2}c_6(\| \mathbf{w}_2 \|^2 + b_2^2) + \frac{1}{2} \| B\mathbf{w}_2 + \mathbf{e}_2 b_2 \|^2 + \frac{c_3}{2}\lambda^T\lambda + \frac{c_4}{2}\theta^T\theta,$$
$$\text{s.a.} \begin{cases} \begin{pmatrix} A\mathbf{w}_2 + \mathbf{e}_1 b_2 \end{pmatrix} + \lambda = \mathbf{e}_1 \\ \begin{pmatrix} C\mathbf{w}_2 + \mathbf{e}_3 b_2 \end{pmatrix} + \theta = \mathbf{e}_3(1-\varepsilon) \end{cases}, \tag{6-3}$$

em que $c_1, c_2, c_3, c_4$ são parâmetros de penalidades, $\lambda \geq 0, \theta \geq 0$ são variáveis de folgas e $\varepsilon$ é um valor entre 0 e 1.

Note que na função perda de (6-2) estamos utilizando norma-2 em relação as variáveis de folgas $\lambda$ e $\theta$ com pesos $c_1/2$ e $c_2/2$ ao invés de norma-1 para $\lambda$ e $\theta$ com pesos $c_1$ e $c_2$ como foi feito em Xu et al. [177], que faz com que as restrições $\lambda \geq 0$ e $\theta \geq 0$ sejam redundantes. Essa simples modificação nos permite resolver os PPQs (6-2) e (6-3) por meio de um sistema de equações lineares.

Realizando uma simples modificação, ou seja, a substituição das restrições de igualdade das Eqs. (6-2) e (6-3), em suas respectivas funções objetivos, modificamos os dois PPQ em um sistema de equações, como mostrado abaixo:

$$\min_{\mathbf{w}_1,b_1} \frac{1}{2}c_5(\| \mathbf{w}_1 \|^2 + b_1^2) + \frac{1}{2} \| A\mathbf{w}_1 + \mathbf{e}_1 b_1 \|^2 + \frac{c_1}{2} \| B\mathbf{w}_1 + \mathbf{e}_2 b_1 + \mathbf{e}_2 \|^2 + \\ + \frac{c_2}{2} \| C\mathbf{w}_1 + \mathbf{e}_3 b_1 + \mathbf{e}_3(1-\varepsilon) \|^2 \tag{6-4}$$

e

$$\min_{\mathbf{w}_2,b_2} \frac{1}{2}c_6(\| \mathbf{w}_2 \|^2 + b_2^2) + \frac{1}{2} \| B\mathbf{w}_2 + \mathbf{e}_2 b_2 \|^2 + \frac{c_3}{2} \| A\mathbf{w}_2 + \mathbf{e}_1 b_2 - \mathbf{e}_1 \|^2 + \\ + \frac{c_4}{2} \| C\mathbf{w}_2 + \mathbf{e}_3 b_2 - \mathbf{e}_3(1-\varepsilon) \|^2 . \tag{6-5}$$

O problema (6-4) pode ser resolvido pelo gradiente em relação a $\mathbf{w}_1$ e $b_1$ igualando a zero, como segue:

$$c_5\mathbf{w}_1 + A^T(A\mathbf{w}_1 + \mathbf{e}_1 b_1) + c_1 B^T(B\mathbf{w}_1 + \mathbf{e}_2 b_1 + \mathbf{e}_2) + c_2 C^T(C\mathbf{w}_1 + \mathbf{e}_3 b_1 + \\ + \mathbf{e}_3(1-\varepsilon)) = 0 \tag{6-6}$$

e

$$c_5 b_1 + \mathbf{e}_1^T(A\mathbf{w}_1 + \mathbf{e}_1 b_1) + c_1 \mathbf{e}_2^T(B\mathbf{w}_1 + \mathbf{e}_2 b_1 + \mathbf{e}_2) + c_2 \mathbf{e}_3^T(C\mathbf{w}_1 + \mathbf{e}_3 b_1 + \\ + \mathbf{e}_3(1-\varepsilon)) = 0, \tag{6-7}$$

combinando as duas expressões (6-6) e (6-7) em forma de matriz temos:

$$c_5\begin{bmatrix}\mathbf{w}_1\\ b_1\end{bmatrix}+\begin{bmatrix}A^TA & A^T\mathbf{e}_1\\ \mathbf{e}_1^TA & \mathbf{e}_1^T\mathbf{e}_1\end{bmatrix}\begin{bmatrix}\mathbf{w}_1\\ b_1\end{bmatrix}+c_1\begin{bmatrix}B^TB & B^T\mathbf{e}_2\\ \mathbf{e}_2^TB & \mathbf{e}_2^T\mathbf{e}_2\end{bmatrix}\begin{bmatrix}\mathbf{w}_1\\ b_1\end{bmatrix}+c_1\begin{bmatrix}B^T\\ \mathbf{e}_2^T\end{bmatrix}\mathbf{e}_2+ \\ +c_2\begin{bmatrix}C^TC & C^T\mathbf{e}_3\\ \mathbf{e}_3^TC & \mathbf{e}_3^T\mathbf{e}_3\end{bmatrix}\begin{bmatrix}\mathbf{w}_1\\ b_1\end{bmatrix}+c_2\begin{bmatrix}C^T\\ \mathbf{e}_3^T\end{bmatrix}\mathbf{e}_3(1-\varepsilon)=0. \tag{6-8}$$

Ajustando a Eq. (6-8), temos:

$$c_5\begin{bmatrix}\mathbf{w}_1\\ b_1\end{bmatrix}+\begin{bmatrix}A^T\\ \mathbf{e}_1^T\end{bmatrix}\begin{bmatrix}A & \mathbf{e}_1\end{bmatrix}\begin{bmatrix}\mathbf{w}_1\\ b_1\end{bmatrix}+c_1\begin{bmatrix}B^T\\ \mathbf{e}_2^T\end{bmatrix}\begin{bmatrix}B & \mathbf{e}_2\end{bmatrix}\begin{bmatrix}\mathbf{w}_1\\ b_1\end{bmatrix}+c_1\begin{bmatrix}B^T\\ \mathbf{e}_2^T\end{bmatrix}\mathbf{e}_2+ \\ +c_2\begin{bmatrix}C^T\\ \mathbf{e}_3^T\end{bmatrix}\begin{bmatrix}C & \mathbf{e}_3\end{bmatrix}\begin{bmatrix}\mathbf{w}_1\\ b_1\end{bmatrix}+c_2\begin{bmatrix}C^T\\ \mathbf{e}_3^T\end{bmatrix}\mathbf{e}_3(1-\varepsilon)=0. \tag{6-9}$$

Iremos simplificar a notação substituindo $\begin{bmatrix}A & \mathbf{e}_1\end{bmatrix}$ por $E$, $\begin{bmatrix}B & \mathbf{e}_2\end{bmatrix}$ por $F$, $\begin{bmatrix}C & \mathbf{e}_3\end{bmatrix}$ por $G$ e $\begin{bmatrix}\mathbf{w}_1\\ b_1\end{bmatrix}$ por $u_1$. Dessa forma, a Eq. (6-9) pode ser reescrita como segue:

$$(c_5I+E^TE+c_1F^TF+c_2G^TG)u_1+c_1F^T\mathbf{e}_2+c_2G^T\mathbf{e}_3(1-\varepsilon)=0. \tag{6-10}$$

Isolando $u_1$ em (6-10) temos:

$$u_1=-\left(c_5I+E^TE+c_1F^TF+c_2G^TG\right)^{-1}\left(c_1F^T\mathbf{e}_2+c_2G^T\mathbf{e}_3(1-\varepsilon)\right). \tag{6-11}$$

De modo análogo ao problema (6-4), podemos calcular $u_2$ para o problema (6-5), obtendo o seguinte:

$$u_2=\left(c_6I+F^TF+c_3E^TE+c_4G^TG\right)^{-1}\left(c_3E^T\mathbf{e}_1+c_4G^T\mathbf{e}_3(1-\varepsilon)\right). \tag{6-12}$$

A regra de decisão para definir à qual classe uma nova amostra pertencerá segue o mesmo método usado para treinar as amostras em uma estrutura "$1-\times-1-\times-$ *restante*", com saídas ternárias $\{-1,0,1\}$. Assim, uma nova amostra $\mathbf{x}_i$, para o caso linear

será rotulada pela função a seguir:

$$f(\mathbf{x}_i) = \begin{cases} +1, & se \quad \mathbf{w}_1^T \mathbf{x}_i + \mathbf{e}b_1 > -1 + \varepsilon \\ -1, & se \quad \mathbf{w}_2^T \mathbf{x}_i + \mathbf{e}b_2 < 1 - \varepsilon \\ 0, & \text{caso contrário.} \end{cases}, \tag{6-13}$$

O rótulo final da classe, para uma nova amostra $\mathbf{x}_i$ é decidido pela regra de votação, onde atribuímos o rótulo "$+1$" para as amostras na primeira classe, rotulamos como "$-1$" para as amostras da segunda classe, e rotulamos por "0" todas as classes restantes. Para o caso linear, o procedimento geral é resumido no Algoritmo (6.2.1).

**Algoritmo 6.2.1** *Dada uma matriz $A \in \mathbb{R}^{m_1 \times n}$ pertencente à classe de rótulos "$+1$", $B \in \mathbb{R}^{m_2 \times n}$ pertencente à classe de rótulos "$-1$" e $C \in \mathbb{R}^{m_3 \times n}$ pertencente à classe das amostras restantes, rotuladas por "0". O caso linear para o algoritmo pode ser obtido usando os seguintes passos:*

*i) Defina $E = [A \quad \boldsymbol{e}_1]$, $F = [B \quad \boldsymbol{e}_2]$, e $G = [C \quad \boldsymbol{e}_3]$;*
*ii) Selecione os parâmetros de penalidade: $\varepsilon$, e $c_i, i = 1, \ldots, 6$;*
*iii) Determine os parâmetros dos dois hiperplanos usando (6-11) e (6-12);*
*iv) Atribua "$+1$","$-1$" ou "0" a uma nova amostra $\boldsymbol{x}_i$ usando a função de decisão (6-13).*

### 6.2.2 Caso não linear

Para estender o caso linear em não linear, consideraremos as seguintes superfícies geradas pelo *kernel* :

$$K(\mathbf{x}^T, D^T)\mathbf{v}_1 + \gamma_1 = 0 \quad e \quad K(\mathbf{x}^T, D^T)\mathbf{v}_2 + \gamma_2 = 0,$$

em que $D = [A; B; C]$, $\mathbf{v}_1$ e $\mathbf{v}_2$ são vetores de pesos, $\gamma_1$ e $\gamma_2$ são *bias*, $K$ é um *kernel* arbitrário. O PPQ primal do caso não linear pode ser modificado de modo semelhante ao realizado em Nasiri et al. [109], com variáveis de folgas utilizando norma-2 e restrições de igualdade substituídas pelas restrições de igualdade como apresentado em (6-14) e (6-15).

$$\begin{aligned} &\min_{\mathbf{v}_1, \gamma_1} \frac{1}{2} c_5 (\| \mathbf{v}_1 \|^2 + \gamma_1^2) + \frac{1}{2} \| K(A, D^T)\mathbf{v}_1 + \mathbf{e}_1 \gamma_1 \|^2 + \frac{c_1}{2} \lambda^T \lambda + \frac{c_2}{2} \theta^T \theta, \\ &\text{s.a.} \begin{cases} -\left( K(B, D^T)\mathbf{v}_1 + \mathbf{e}_2 \gamma_1 \right) + \lambda = \mathbf{e}_2 \\ -\left( K(C, D^T)\mathbf{v}_1 + \mathbf{e}_3 \gamma_1 \right) + \theta = \mathbf{e}_3 (1 - \varepsilon) \end{cases}, \end{aligned} \tag{6-14}$$

e

$$\min_{\mathbf{v}_2,\gamma_2} \frac{1}{2}c_6(\| \mathbf{v}_2 \|^2 + \gamma_2^2) + \frac{1}{2} \| K(B,D^T)\mathbf{v}_2 + \mathbf{e}_2\gamma_2 \|^2 + \frac{c_3}{2}\lambda^T\lambda + \frac{c_4}{2}\theta^T\theta,$$
$$\text{s.a.} \begin{cases} \left(K(A,D^T)\mathbf{v}_2 + \mathbf{e}_1\gamma_2\right) + \lambda = \mathbf{e}_1 \\ \left(K(C,D^T)\mathbf{v}_2 + \mathbf{e}_3\gamma_2\right) + \theta = \mathbf{e}_3(1-\varepsilon) \end{cases} . \tag{6-15}$$

Pela substituições das restrições na função objetivo, os PPQs se tornam:

$$\begin{aligned} \min_{\mathbf{v}_1,\gamma_1} \quad & \frac{1}{2}c_5(\| \mathbf{v}_1 \|^2 + \gamma_1^2) + \frac{1}{2} \| K(A,D^T)\mathbf{v}_1 + \mathbf{e}_1\gamma_1 \|^2 + \frac{c_1}{2} \| K(B,D^T)\mathbf{v}_1 + \\ & + \mathbf{e}_2\gamma_1 + \mathbf{e}_2 \|^2 + \frac{c_2}{2} \| K(C,D^T)\mathbf{v}_1 + \mathbf{e}_3\gamma_1 + \mathbf{e}_3(1-\varepsilon) \|^2 \end{aligned} \tag{6-16}$$

e

$$\begin{aligned} \min_{\mathbf{v}_2,\gamma_2} \quad & \frac{1}{2}c_6(\| \mathbf{v}_2 \|^2 + \gamma_2^2) + \frac{1}{2} \| K(B,D^T)\mathbf{v}_2 + \mathbf{e}_2\gamma_2 \|^2 + \frac{c_3}{2} \| K(A,D^T)\mathbf{v}_2 + \\ & + \mathbf{e}_1\gamma_2 - \mathbf{e}_1 \|^2 + \frac{c_4}{2} \| K(C,D^T)\mathbf{v}_2 + \mathbf{e}_3\gamma_2 - \mathbf{e}_3(1-\varepsilon) \|^2 . \end{aligned} \tag{6-17}$$

A solução do PPQ (6-16) pode ser resolvido por meio do gradiente, em relação a $\mathbf{v}_1$ e $\gamma_1$ igualando a zero como segue:

$$\begin{aligned} & c_5\mathbf{v}_1 + K(A,D^T)^T\left(K(A,D^T)\mathbf{v}_1 + \mathbf{e}_1\gamma_1\right) + c_1K(B,D^T)^T\left(K(B,D^T)\mathbf{v}_1 + \right. \\ & \left. + \mathbf{e}_2\gamma_1 + \mathbf{e}_2\right) + c_2K(C,D^T)^T(K(C,D^T)\mathbf{v}_1 + \mathbf{e}_3\gamma_1 + \mathbf{e}_3(1-\varepsilon)) = 0 \end{aligned} \tag{6-18}$$

e

$$\begin{aligned} & c_5\gamma_1 + \mathbf{e}_1^T\left(K(A,D^T)\mathbf{v}_1 + \mathbf{e}_1\gamma_1\right) + c_1\mathbf{e}_2^T\left(K(B,D^T)\mathbf{v}_1 + \mathbf{e}_2\gamma_1 + \mathbf{e}_2\right) + \\ & c_2\mathbf{e}_3^T\left(K(C,D^T)\mathbf{v}_1 + \mathbf{e}_3\gamma_1 + \mathbf{e}_3(1-\varepsilon)\right) = 0. \end{aligned} \tag{6-19}$$

Trabalhando com (6-18) e (6-19), obtemos o seguinte:

$$
\begin{aligned}
&c_5\begin{bmatrix}\mathbf{v}_1\\ \gamma_1\end{bmatrix}+\begin{bmatrix}K(A,D^T)^T\\ \mathbf{e}_1^T\end{bmatrix}\begin{bmatrix}K(A,D^T) & \mathbf{e}_1\end{bmatrix}\begin{bmatrix}\mathbf{v}_1\\ \gamma_1\end{bmatrix}+c_1\begin{bmatrix}K(B,D^T)^T\\ \mathbf{e}_2^T\end{bmatrix}\\
&\begin{bmatrix}K(B,D^T) & \mathbf{e}_2\end{bmatrix}\begin{bmatrix}\mathbf{v}_1\\ \gamma_1\end{bmatrix}+c_1\begin{bmatrix}K(B,D^T)^T\\ \mathbf{e}_2^T\end{bmatrix}\mathbf{e}_2+c_2\begin{bmatrix}K(C,D^T)^T\\ \mathbf{e}_3^T\end{bmatrix}\begin{bmatrix}K(C,D^T) & \mathbf{e}_3\end{bmatrix} \qquad (6\text{-}20)\\
&\begin{bmatrix}\mathbf{v}_1\\ \gamma_1\end{bmatrix}+c_2\begin{bmatrix}K(C,D^T)^T\\ \mathbf{e}_3^T\end{bmatrix}\mathbf{e}_3(1-\varepsilon)=0,
\end{aligned}
$$

onde, iremos renomear alguns termos para facilitar a escrita, $\begin{bmatrix}K(A,D^T) & \mathbf{e}_1\end{bmatrix}$ por $M$, $\begin{bmatrix}K(B,D^T) & \mathbf{e}_2\end{bmatrix}$ por $N$, $\begin{bmatrix}K(C,D^T) & \mathbf{e}_3\end{bmatrix}$ por $O$ e $\begin{bmatrix}\mathbf{v}_1\\ \gamma_1\end{bmatrix}$ por $\varphi_1$, desta forma, podemos rescrever (6-20) como segue:

$$(c_5I+M^TM+c_1N^TN+c_2O^TO)\varphi_1+c_1N^T\mathbf{e}_2+c_2O^T\mathbf{e}_3(1-\varepsilon)=0. \qquad (6\text{-}21)$$

Isolando $\varphi_1$ em (6-21) temos:

$$\varphi_1=-\Big(c_5I+M^TM+c_1N^TN+c_2O^TO\Big)^{-1}\Big(c_1N^T\mathbf{e}_2+c_2O^T\mathbf{e}_3(1-\varepsilon)\Big). \qquad (6\text{-}22)$$

Seguindo o procedimento, de modo análogo à solução do problema (6-16), encontramos $\varphi_2$ para o problema (6-17) como apresentado a seguir:

$$\varphi_2=\Big(c_6I+N^TN+c_3M^TM+c_4O^TO\Big)^{-1}\Big(c_3M^T\mathbf{e}_1+c_4O^T\mathbf{e}_3(1-\varepsilon)\Big). \qquad (6\text{-}23)$$

Neste ponto, podemos notar que a solução para o caso não linear exige inversões de matrizes com dimensões $(m+1)\times(m+1)$. A fórmula de Sherman-Morrison-Woodbury (SMW) é amplamente utilizada em várias áreas de pesquisa, tais como: estatística, otimização, análise assintótica e equações diferenciais parciais [174]. Considere $A\in\mathbb{R}^{n\times n}, U\in\mathbb{R}^{n\times r}$ e $V\in\mathbb{R}^{n\times r}$. Se $A$ e $(I+V^TA^{-1}U)$ são não singulares, então, $(A+UV^T)$ também é não singular, a fórmula (6-24) de Sherman-Morrison-Woodbury é apresentada a seguir:

$$(A+UV^T)^{-1}=A^{-1}-A^{-1}U(I+V^TA^{-1}U)^{-1}V^TA^{-1}. \qquad (6\text{-}24)$$

A fórmula SMW fornece uma maneira eficaz de calcular $(A+UV^T)^{-1}$. Assim, com

o intuito de reduzir o custo computacional, a fórmula de Sherman-Morrison-Woodbury (SMW) [47] foi usada nas equações (6-22) e (6-23). Seguindo Kumar e Gopal [79] e a ideia de Nasiri et al. [109], mostramos que o caso não linear pode ser resolvido calculando cinco inversões de matrizes de menores dimensões que $(m+1)\times(m+1)$. A seguir serão discutidas as soluções para o caso não linear.

Caso 1: $m_1 \leq m_2$

Usando a fórmula de SMW, podemos reescrever (6-22) e (6-23) como segue:

$$\varphi_1 = -\left(X_1 - X_1 M^T \left(I + M X_1 M^T\right)^{-1} M X_1\right)\left(c_1 N^T \mathbf{e}_2 + c_2 O^T \mathbf{e}_3 (1-\varepsilon)\right) \tag{6-25}$$

e

$$\varphi_2 = \left(X_2 - X_2 M^T \left(I/c_3 + M X_2 M^T\right)^{-1} M X_2\right)\left(c_3 M^T \mathbf{e}_1 + c_4 O^T \mathbf{e}_3 (1-\varepsilon)\right), \tag{6-26}$$

em que $X_1 = \left(c_5 I + M^T M + c_1 N^T N + c_2 O^T O\right)^{-1}$ e $X_2 = \left(c_6 I + N^T N + c_3 M^T M + c_4 O^T O\right)^{-1}$ pode ser encontrada usando a fórmula SMW do seguinte modo:

$$X_1 = \frac{1}{c_1}\left[Y_1 - Y_1 N^T \left(I + N Y_1 N^T\right)^{-1} N Y_1\right] \tag{6-27}$$

e

$$X_2 = \left[Y_2 - Y_2 N^T \left(I + N Y_2 N^T\right)^{-1} N Y_2\right], \tag{6-28}$$

novamente usando a fórmula SMW, temos:

$$Y_1 = \frac{c_1}{c_2}\left[Z - Z\left(\frac{c_2}{c_5} I + Z\right)^{-1} Z\right] \tag{6-29}$$

e

$$Y_2 = \frac{1}{c_4}\left[Z - Z\left(\frac{c_4}{c_6} I + Z\right)^{-1} Z\right]. \tag{6-30}$$

Para calcular $Y_1$ e $Y_2$ precisamos calcular inicialmente o valor de $Z = \left(O^T O\right)^{-1}$. Inspirado em Jayadeva et al. [66], introduzimos o termo de regularização $\alpha I$, $\alpha > 0$ em $Z$ para resolver o problema do mal condicionamento da matriz, ou seja, caso a matriz $(O^T O)^{-1}$ seja singular,

$$Z = \frac{1}{\alpha}\left(I - O^T \left(\alpha I + O O^T\right)^{-1} O\right). \tag{6-31}$$

Caso 2: $m_1 > m_2$

Similarmente ao que foi feito nas Eqs. (6-25) e (6-30), foram obtidas as equações para o caso $m_1 > m_2$ como segue:

$$\varphi_1 = -\left(X_3 - X_3N^T\left(\frac{I}{c_1} + NX_3N^T\right)^{-1}NX_3\right)\left(c_1N^T\mathbf{e}_2 + c_2O^T\mathbf{e}_3(1-\varepsilon)\right) \tag{6-32}$$

e

$$\varphi_2 = \left(X_4 - X_4N^T\left(I + NX_4N^T\right)^{-1}NX_4\right)\left(c_3M^T\mathbf{e}_1 + c_4O^T\mathbf{e}_3(1-\varepsilon)\right), \tag{6-33}$$

em que $X_3 = \left(c_5I + M^TM + c_1N^TN + c_2O^TO\right)^{-1}$ e $X_4 = \Big(c_6I + N^TN + c_3M^TM + c_4O^TO\Big)^{-1}$ pode ser encontrado utilizando a fórmula de SMW como segue:

$$X_3 = \left[Y_3 - Y_3M^T\left(I + MY_3M^T\right)^{-1}MY_3\right] \tag{6-34}$$

e

$$X_4 = \frac{1}{c_3}\left[Y_4 - Y_4M^T\left(I + MY_4M^T\right)^{-1}MY_4\right], \tag{6-35}$$

novamente usando a fórmula SMW obtemos:

$$Y_3 = \frac{1}{c_2}\left[Z - Z\left(\frac{c_2}{c_5}I + Z\right)^{-1}Z\right] \tag{6-36}$$

e

$$Y_4 = \frac{c_3}{c_4}\left[Z - Z\left(\frac{c_4}{c_6}I + Z\right)^{-1}Z\right]. \tag{6-37}$$

$Z$ é o mesmo obtido pela Eq. (6-31).

Desta maneira, uma nova amostra $\mathbf{x}_i$ será rotulada pela função de decisão conforme apresentado a seguir:

$$f(\mathbf{x}_i) = \begin{cases} +1, & se \quad K(\mathbf{x}_i, D^T)\mathbf{v}_1 + e\gamma_1 > -1 + \varepsilon \\ -1, & se \quad K(\mathbf{x}_i, D^T)\mathbf{v}_2 + e\gamma_2 < 1 - \varepsilon \\ 0, & \text{caso contrário} \end{cases}. \tag{6-38}$$

A classe final, análoga ao realizado para o caso linear, é atribuída para a classe de rótulos que obteve mais votos. Para o caso não linear, o procedimento geral para calcular em qual classe uma nova amostra pertencerá é resumido no Algoritmo (6.2.2).

**Algoritmo 6.2.2** *Dada uma matriz $A \in \mathbb{R}^{m_1 \times n}$ pertencente à classe de rótulos "$+1$",*

*$B \in \mathbb{R}^{m_2 \times n}$ pertencente à classe de rótulos "$-1$", $C \in \mathbb{R}^{m_3 \times n}$ sendo o restante das amostras, pertencentes à classe de rótulos "0" e $D = [A;B;C]$. O caso não linear para o algoritmo proposto é obtido usando os seguintes passos:*

*i) Escolha a função kernel K que será usada;*

*ii) Defina $D = [A;B;C]$, $M = [K(A,D^T) \quad \boldsymbol{e}_1]$, $N = [K(B,D^T) \quad \boldsymbol{e}_2]$ e $O = [K(C,D^T) \quad \boldsymbol{e}_3]$;*

*iii) Selecione os parâmetros de penalidades: $\alpha$, $\varepsilon$ e $c_i, i = 1,\ldots,6$;*

*iv) Se $(m_1 \leq m_2)$*
*determine os parâmetros das duas hiper-superfícies usando (6-25)↔(6-31);*

*v) Se $(m_1 > m_2)$*
*determine os parâmetros das duas hiper-superfícies usando (6-32)↔(6-37) e (6-31);*

*vi) Atribua "$+1$","$-1$" ou "0" a uma nova amostra $\boldsymbol{x}_i$ usando a função de decisão (6-38).*

## 6.3 Experimentos Numéricos

Para avaliar o desempenho do algoritmo proposto, nesta seção será investigado a sua performance em dez bases de dados extraídas do repositório da UCI [36], conforme descrito na Tabela (6.1). Todas as amostra foram normalizadas entre $[0,1]$ antes da aplicação dos algoritmos. Os testes foram realizados comparando três algoritmos: Twin-KSVC, LSTKSVC e o algoritmo proposto.

Foi usada a função "*quadprog.m*" do *Matlab* para solucionar o PPQ do Twin-KSVC. Em todos os experimentos foi utilizado o método de validação cruzada *k-fold*, com $k = 5$. Esse processo consiste em dividir aleatoriamente $k$ subconjuntos, sendo que um desses subconjuntos é reservado para o conjunto de testes. Esse processo é repetido $k$ vezes, e a média das acurácias dos $k$ resultados de testes é usado como uma medida de desempenho.

**Tabela 6.1:** *Bases de dados usados nos experimentos.*

| Bases de dados | Número de amostras | Número de variáveis | Número de classes |
|---|---|---|---|
| *Balance* | 625 | 4 | 3 |
| *Ecoli* | 327 | 7 | 5 |
| *Zoo* | 101 | 16 | 7 |
| *Iris* | 150 | 4 | 3 |
| *Seeds* | 210 | 7 | 3 |
| *Thyroid* | 215 | 5 | 3 |
| *Wine* | 178 | 13 | 3 |
| *Soybean* | 47 | 35 | 4 |
| *Dermatology* | 358 | 34 | 6 |
| *Vowel* | 990 | 10 | 11 |

### 6.3.1 Seleção de Parâmetros

Um problema preliminar, tanto para o TWSVM e suas extensões, é a escolha de ótimos parâmetros. Neste experimento foi selecionada a função *kernel Gaussiano* $K(\mathbf{x}_i,\mathbf{x}_j) = exp(-\parallel \mathbf{x}_i-\mathbf{x}_j \parallel^2)/\gamma^2$ e um método de busca em grade para obter os melhores parâmetros. Para o algoritmo Twin-KSVC, o parâmetro $\gamma$ para o *kernel Gaussiano* foi selecionado no intervalo $\{2^i \mid i = -6,-5,\ldots,6\}$. Os parâmetros $c_1$, $c_2$, $c_3$ e $c_4$ foram selecionados no intervalo de $\{2^i \mid i = -10,-9,\ldots,6\}$ e para reduzir a complexidade computacional, utilizamos $c_1 = c_3$ e $c_2 = c_4$. O parâmetro $\varepsilon$ foi escolhido no conjunto $\{0,0.1,0.2,0.3\}$.

Para o algoritmo proposto e o algoritmo LSTKSVC, os ótimos valores para o parâmetro $c_i$ e $\gamma$ foram selecionados no intervalo $\{2^i \mid i = -8,-7,\ldots,8\}$. Com o intuito de reduzir a complexidade computacional dos algoritmos, os parâmetros foram usados do seguinte modo: $c_1 = c_3$, $c_2 = c_4$ em ambos algoritmos e $c_5 = c_6$ usado no algoritmo proposto, agindo como um fator peso. O fator peso determina o *tradeoff* entre o termo de regularização e o risco empírico [146]. O parâmetro $\varepsilon$ foi escolhido no conjunto $\{0.1,0.2,0.3,0.4\}$.

### 6.3.2 Resultados e discussões

Na Tabela (6.2) mostramos os resultados do desempenho dos algoritmos e os parâmetros usados para obter os melhores resultados. A acurácia (Acc) denota a média das acurácias obtidas das 5 vezes em que os testes foram avaliados, seguida de mais ou menos o valor médio do desvio padrão obtido. O tempo é a média do tempos obtidos, em segundos, para avaliar os 5 experimentos, considerando o tempo de treinamento e teste.

**Tabela 6.2:** *Comparação de desempenho dos algoritmos de múltiplas classes utilizando kernel RBF.*

| Bases de dados | Twin-KSVC<br>Acc±Desvio padrão<br>$(c_1, c_2, \gamma, \varepsilon)$<br>Tempo (s) | LSTKSVC<br>Acc±Desvio padrão<br>$(c_1, c_2, \gamma, \varepsilon)$<br>Tempo (s) | Algoritmo Proposto<br>Acc±Desvio padrão<br>$(c_1, c_2, c_5, \gamma, \varepsilon)$<br>Tempo (s) |
|---|---|---|---|
| *Balance* | **92,81 ± 2,80**<br>$(2^{-7}, 2^{-4}, 2^{-4}, 0.2)$<br>12,99 | 91,20 ± 1,87<br>$(2^{5}, 2^{-3}, 2^{-2}, 0.3)$<br>15,93 | 90,88 ± 0,71<br>$(2^{3}, 2^{-3}, 2^{-1}, 2^{-1}, 0.2)$<br>17,26 |
| *Ecoli* | 79,46 ± 4,65<br>$(2^{0}, 2^{-2}, 2^{4}, 0.1)$<br>39,62 | 83,34 ± 3,84<br>$(2^{3}, 2^{-8}, 2^{8}, 0.2)$<br>44,52 | **84,52± 3,10**<br>$(2^{3}, 2^{-8}, 2^{-8}, 2^{8}, 0.2)$<br>44,51 |
| *Zoo* | 88,03 ± 10,23<br>$(2^{-5}, 2^{-7}, 2^{2}, 0.2)$<br>2,85 | **97,04 ± 4,32**<br>$(2^{7}, 2^{-7}, 2^{2}, 0.3)$<br>4,32 | 96,09 ± 6,23<br>$(2^{8}, 2^{-8}, 2^{8}, 2^{-8}, 0.2)$<br>3,05 |
| *Iris* | 94,00 ± 4,35<br>$(2^{-7}, 2^{-4}, 2^{0}, 0.0)$<br>0,78 | **96,00 ± 3,65**<br>$(2^{3}, 2^{-3}, 2^{4}, 0.3)$<br>0,96 | **96,00 ± 4,34**<br>$(2^{6}, 2^{-8}, 2^{0}, 2^{0}, 0.1)$<br>0,91 |
| *Seeds* | 90,48 ± 3,37<br>$(2^{-7}, 2^{-4}, 2^{2}, 0.2)$<br>1,53 | 94,76 ± 3,10<br>$(2^{1}, 2^{-6}, 2^{-2}, 0.3)$<br>1,75 | **95,71± 1,06**<br>$(2^{3}, 2^{-6}, 2^{-8}, 2^{2}, 0.2)$<br>1,92 |
| *Thyroid* | 92,09 ± 3,12<br>$(2^{-7}, 2^{-4}, 2^{0}, 0.2)$<br>1,59 | 95,35 ± 4,03<br>$(2^{1}, 2^{-8}, 2^{-2}, 0.3)$<br>1,85 | **97,21 ± 1,95**<br>$(2^{2}, 2^{-6}, 2^{2}, 2^{-7}, 0.3)$<br>2,02 |
| *Wine* | 94,31 ± 3,67<br>$(2^{-5}, 2^{-4}, 2^{2}, 0.0)$<br>1,10 | 97,76 ± 2,33<br>$(2^{4}, 2^{-6}, 2^{8}, 0.3)$<br>2,33 | **99,43±1,28**<br>$(2^{5}, 2^{-6}, 2^{-6}, 2^{3}, 0.2)$<br>1,38 |
| *Soybean* | **100 ± 0**<br>$(2^{-10}, 2^{-8}, 2^{-6}, 0.3)$<br>0,29 | **100,0 ± 0**<br>$(2^{-1}, 2^{-6}, 2^{-1}, 0.3)$<br>0,19 | **100,0± 0**<br>$(2^{-8}, 2^{-8}, 2^{-7}, 2^{-7}, 0.1)$<br>0,27 |
| *Dermatology* | 84,47 ± 5,39<br>$(2^{-2}, 2^{-6}, 2^{4}, 0.3)$<br>24,61 | 90,73 ± 2,55<br>$(2^{7}, 2^{-3}, 2^{7}, 0.1)$<br>26,89 | **93,46± 2,21**<br>$(2^{8}, 2^{-8}, 2^{-8}, 2^{7}, 0.1)$<br>29,05 |
| *Vowel* | 89,09 ± 0,002<br>$(2^{-2}, 2^{-7}, 2^{4}, 0.3)$<br>23,73 | 94,95 ± 1,38<br>$(2^{-1}, 2^{-6}, 2^{-1}, 0.3)$<br>22,67 | **98,28 ± 1,10**<br>$(2^{1}, 2^{-7}, 2^{-1}, 2^{-1}, 0.3)$<br>22,62 |
| Média das acurácias | 90,47 ± 3,76 | 94,11 ± 2,70 | **95,16 ± 2,43** |

Os valores em negrito, apresentados na Tabela (6.2), indicam os melhores resultados dos experimentos (em %). Da perspectiva dos testes para obtenção da acurácia, podemos notar que o algoritmo proposto superou os outros dois algoritmos em quase todos os bancos de dados usados. A acurácia média do algoritmo proposto nas dez bases de dados foi de $95,16\%$, comparado com $90,47\%$ e $94,11\%$ para o Twin-KSVC e LSTKSVC, respectivamente. A principal razão dessa eficiência é que o algoritmo proposto, além de implementar o princípio de minimização do risco estrutural ao introduzir o termo de regularização, reduz a solução de dois problemas primais modificados, resolvendo dois sistemas de equações lineares, ao contrário do Twin-KSVC que necessita resolver dois PPQs. Os três algoritmos obtiveram comparável desempenho na base de dados *Soybean*. Contudo, podemos observar que a acurácia obtida pelo algoritmo proposto é melhor do que outros métodos aplicados às seguintes bases de dados: *Ecoli, Seeds, Thyroid, Wine, Dermatology* e *Vowel*, sendo que na base de dados *Iris* o resultado obtido foi igual ao do algoritmo LSTKSVC.

### 6.3.3 Análise estatística

Com o propósito de analisar ainda mais o desempenho dos três algoritmos, em relação aos 10 bancos de dados, por meio de um método estatístico, utilizamos o teste de Friedman [59] com os correspondentes testes *post hoc*, como sugerido por Demšar [35], e é comumente utilizado por vários pesquisadores [150, 175, 37, 168].

O algoritmo proposto supera os outros dois algoritmos em quase todas as bases de dados, em termos de acurácia, mostrados na Tabela (6.2). Porém, em duas bases de dados a acurácia é levemente menor do que os outros dois algoritmos. O teste de Friedman ordena os algoritmos em cada conjunto de dados separadamente, sendo que o algoritmo com o melhor desempenho obtém o valor 1 no ranking, o segundo algoritmo recebe o valor 2 no ranking e assim por diante. Em caso de empate, a média do ranking é atribuído a cada um dos algoritmos.

Sob a hipótese nula de que todos os algoritmos são equivalentes, a estatística do teste de Friedman Eq. (6-39) é distribuída de acordo com $\chi_F^2$ com $(k-1)$ graus de liberdade, como apresentado a seguir:

$$\chi_F^2 = \frac{12N}{k(k+1)}\left[\sum_j R_j^2 - \frac{k(k+1)^2}{4}\right], \tag{6-39}$$

em que $R_j = \frac{1}{N}\sum_j r_i^j$, e $r_i^j$ denota o $j$-ésimo dos $k$ algoritmos e $i$-ésimo dos $N$ conjuntos de dados.

O cálculo do $\chi_F^2$ do teste de Friedman é mostrado na Eq. (6-40).

$$F_f = \frac{(N-1)\chi_F^2}{N(k-1)-\chi_F^2}. \tag{6-40}$$

A Tabela (6.3) mostra o ranking de cada algoritmo de classificação, de acordo com a acurácia obtida para cada conjunto de dados, utilizando o *kernel* RBF. Os valores em negritos na Tabela (6.3) retrata o ranking do melhor classificador.

**Tabela 6.3:** *Ranking dos algoritmos de classificação: Twin-KSCV, LSTKSVC e o algoritmo proposto, utilizando kernel RBF.*

| Bases de dados | Twin-KSVC | LSTKSVC | Improvements on LSTKSVC |
|---|---|---|---|
| *Balance* | 1 | 2 | 3 |
| *Ecoli* | 3 | 2 | 1 |
| *Zoo* | 3 | 1 | 2 |
| *Iris* | 3 | 1,5 | 1,5 |
| *Seeds* | 3 | 2 | 1 |
| *Thyroid* | 3 | 2 | 1 |
| *Wine* | 3 | 2 | 1 |
| *Soybean* | 2 | 2 | 2 |
| *Dermatology* | 3 | 2 | 1 |
| *Vowel* | 3 | 2 | 1 |
| Ranking médio | 2,7 | 1,85 | **1,45** |

De acordo com as Eqs. (6-39) e (6-40) obtemos $\chi_F^2$ e $F_f$ como segue:

$$\chi_F^2 = \frac{12\times 10}{3(3+1)}\left[2,7^2+1,85^2+1,45^2-\frac{3(3+1)^2}{4}\right] = 8,15$$

e

$$F_f = \frac{(10-1)\times 8,15}{10(3-1)-8,15} = 6,189.$$

Com os três algoritmos e dez bases de dados, $F_f$ é distribuído de acordo com a distribuição $F$ com $(k-1)$ e $(k-1)(N-1) = (2,18)$ graus de liberdade. Os valores críticos de $F(2,18)$ para $\alpha = 0,05$ é 3,56, e de modo similar 6,01 para $\alpha = 0,01$. Uma vez que o valor de $F_f$ é muito maior do que os valores críticos, há uma significante diferença entre os três algoritmos. Note que o ranking médio do algoritmo proposto é menor do que o restante dos outros algoritmos. Isso significa que o algoritmo proposto é mais preciso que os outros dois algoritmos.

Realizamos uma comparação com outros estudos que utilizaram bases de dados com múltiplas classes, confirmando ainda mais o desempenho do nosso algoritmo. Os

resultados obtidos pelos vários métodos são apresentados na Tabela (6.4). Os resultados das acurácias foram extraídos diretamente das literaturas referenciadas. A comparação é baseada nos resultados obtidos em cada estudo individual. Os resultados em negritos são os que apresentam maiores acurácias. Nas sete bases de dados analisadas, nosso algoritmo obteve desempenho maior em quatro delas (*Glass, Contraceptive, Car* e *Vehicle*), em outras duas bases (*Hoyes-Roth* e *Libras*) obteve desempenho suavemente menor, porém muito próximo aos outros algoritmos.

**Tabela 6.4:** *Comparativo entre as acurácias obtidas entre vários algoritmos encontrados na literatura.*

| Bases de dados | Dimensão | Algoritmos | Acurácias (%) | Referências |
|---|---|---|---|---|
| *Teaching Evaluation* | 151 × 5 × 3 | Twin-KSVC | **71,01±23,07** | [177] |
| | | PNPSVM(II) | 69,92 ±5,24 | [60] |
| | | Nosso Estudo | 58,99±6,65 | |
| *Glass* | 214 × 9 × 6 | IMBSVM | 64,01±3,65 | [187] |
| | | PNPSVM(II) | 66,24±3,38 | [72] |
| | | F-LS-TWSVC | 65,41±5,52 | [131] |
| | | Nosso Estudo | **67,73±5,39** | |
| *Hoyes-Roth* | 132 × 5 × 3 | Twin-KSVC | 81,10±3,95 | [177] |
| | | PNPSVM(II) | **81,65±4,16** | [60] |
| | | Nosso Estudo | 79,54±7,40 | |
| *Contraceptive* | 1473 × 9 × 3 | WMGEPSVM | 53,30±3,20 | [76] |
| | | MLSNHSVM | 44,00±5,70 | [77] |
| | | MLSTSVM | 50,62±4,66 | [156] |
| | | Nosso Estudo | **53,84±3,13** | |
| *Car* | 1728 × 6 × 3 | Twin-KSVC | 72,30±5,04 | [177] |
| | | LST-KSVC | 94,13±0,17 | [109] |
| | | Nosso Estudo | **94,92±0,82** | |
| *Vehicle* | 846 × 18 × 4 | RFLSTSVM-CIL | 69,86±7,57 | [132] |
| | | TPSVM-L1 | 51,22± [1] | [49] |
| | | DTTSVM | 77,37±3,17 | [140] |
| | | Nosso Estudo | **78,60±2,69** | |
| *Libras* | 360 × 90 × 15 | MRADAG | **72,56±3,20** | [148] |
| | | Nosso Estudo | 72,48±4,12 | |

[1] Valor do desvio padrão não fornecido no artigo original.

## 6.4 Comentários Finais

Nesse capítulo, propomos um novo algoritmo, que atua como uma melhoria na versão mínimos quadrados do LSTKSVC. Nesse algoritmo é implementado o princípio de minimização do risco estrutural, por introduzir um termo de regularização na versão do LSTKSVC para problemas de múltiplas classes, o que ajuda a melhorar a habilidade de generalização. O algoritmo proposto avalia as amostras dos dados de treinamento em uma estrutura "$1-\times-1-\times-$ *restante*", gerando saídas ternárias $\{-1,0,+1\}$. Diferentemente do Twin-KSVC que precisa resolver dois PPQ, o algoritmo proposto resolve dois problemas primais ao resolver um sistema de equações lineares semelhante ao LSTKSVC. Os resultados dos experimentos sobre os bancos de dados da *UCI* mostraram a efetividade da acurácia de classificação do algoritmo proposto. Esse estudo resultou em um artigo publicado no *Journal Neurocomputing* com o título: *Improvements on least squares twin multi-class classification support vector machine* [90].

CAPÍTULO 7

# Algumas aplicações

## 7.1 Considerações Iniciais

Neste capítulo, serão apresentadas as aplicações, sendo a primeira relativa à classificação de amostras de laranjas cultivadas em dois sistemas, o orgânico e o convencional, a segunda em relação ao reconhecimento geográfico de vinhos produzidos em dois países (Argentina e Chile), a terceira em relação à classificação de vegetais (alface, cebola, tomate e pimentão) quanto ao sistema de cultivo (orgânico e convencional) e a quarta elaboramos um seletor de variáveis e utilizamos bases de dados médicos, disponíveis no repositório da UCI Irvine, para mostrar sua eficácia. Esses trabalhos culminaram na publicação de dois artigos [162] e [32] e dois artigos submetidos.

## 7.2 Classificação de amostras de laranjas cultivadas nos sistemas orgânico e convencional, utilizando os elementos químicos obtidos pelo ICP-MS

### 7.2.1 Introdução

A laranja é uma fruta cítrica que é muito consumida em todo o mundo. Laranjeiras podem ser encontradas em vários países, mas produção em larga escala ocorre em apenas alguns. O Brasil é um dos maiores produtores mundiais de laranja, e o principal exportador de suco de laranja [161]. O tipo de sistema de cultivo, seja convencional ou orgânico, afeta a composição dos frutos, portanto, análise e estudos do nível de determinados elementos químicos e nutrientes fornecem informações valiosas, como apresentado por Tarozzi et al. [151].

O objetivo deste projeto é usar técnicas de mineração de dados, chamadas máquinas de vetores de suporte (SVM) e perceptron múltiplas camadas (MLP), combinada com a seleção de variáveis, tais como: *F-score* e $\chi^2$, para classificar amostras de laranjas cultivadas orgânica e convencionalmente, com base em seus elementos químicos e forne-

cer um modelo de classificação que prevê se uma determinada amostra de laranja foi ou não cultivada em um sistema orgânico.

### 7.2.2 Conjunto de dados analisado

Foram analisadas 57 amostras de laranja (29 cultivadas convencionalmente e 28 cultivadas organicamente). As amostras de laranjas foram obtidas em Borborema no estado de São Paulo. Foram analisadas laranjas da variedade valência (*Citrus sinensis (L.) Osbeck*) budded on Rangpur lime (*Citrus limonia Osbeck*).

### 7.2.3 Metodologia

As análises das amostras de laranjas foram realizadas utilizando *inductively coupled plasma-mass spectrometry* (ICP-MS), quantificando os elementos químicos encontrados nas amostras (Al, Cu, Mn, Rb, Sr, Cs, Ba, Zn, Fe, Si e Mg). Para classificação dessas amostras, algoritmos de mineração de dados foram utilizados (SVM e MLP), combinados com seleção de variáveis (*F-score* e $\chi^2$).

Foram obtidos os valores médios, mínimos e máximos de concentração para os elementos químicos determinados. As amostras foram normalizadas entre $[0,1]$. Logo após foram utilizados os algoritmos seletores de variáveis para obter os subconjuntos de variáveis para serem utilizados com os algoritmos de mineração de dados. Os testes foram realizados utilizando o método de validação cruzada (10-*fold cross validation*).

### 7.2.4 Resultados

A Tabela (7.1) mostra as combinações obtidas usando o *F-score*, onde o subconjunto #$F01$ é o subconjunto da variável com o melhor *F-score*, o subconjunto #$F02$ é o subconjunto formado pelas duas variáveis com os dois melhores resultados, e assim por diante até o subconjunto #$F11$ que contém todas as variáveis. De modo análogo, foi realizada a composição da Tabela (7.2), utilizando o $\chi^2$.

**Tabela 7.1:** *Subconjuntos baseado no ranking do F-score.*

| Subconjunto | Variáveis |
|---|---|
| #F01 | "Si" |
| #F02 | "Si, Cu" |
| #F03 | "Si, Cu, Cs" |
| #F04 | "Si, Cu, Cs, Mn" |
| #F05 | "Si, Cu, Cs, Mn, Fe" |
| #F06 | "Si, Cu, Cs, Mn, Fe, Zn" |
| #F07 | "Si, Cu, Cs, Mn, Fe, Zn, Sr" |
| #F08 | "Si, Cu, Cs, Mn, Fe, Zn, Sr, Al" |
| #F09 | "Si, Cu, Cs, Mn, Fe, Zn, Sr, Al, Ba" |
| #F10 | "Si, Cu, Cs, Mn, Fe, Zn, Sr, Al, Ba, Rb" |
| #F11 | "Si, Cu, Cs, Mn, Fe, Zn, Sr, Al, Ba, Rb, Mg" |

**Tabela 7.2:** *Subconjuntos baseado no ranking do* $\chi^2$.

| Subconjunto | Variáveis |
|---|---|
| #X01 | "Cu" |
| #X02 | "Cu, Cs" |
| #X03 | "Cu, Cs, Zn" |
| #X04 | "Cu, Cs, Zn, Al" |
| #X05 | "Cu, Cs, Zn, Al, Mn" |
| #X06 | "Cu, Cs, Zn, Al, Mn, Rb" |
| #X07 | "Cu, Cs, Zn, Al, Mn, Rb, Sr" |
| #X08 | "Cu, Cs, Zn, Al, Mn, Rb, Sr, Ba" |
| #X09 | "Cu, Cs, Zn, Al, Mn, Rb, Sr, Ba, Fe" |
| #X10 | "Cu, Cs, Zn, Al, Mn, Rb, Sr, Ba, Fe, Mg" |
| #X11 | "Cu, Cs, Zn, Al, Mn, Rb, Sr, Ba, Fe, Mg, Si" |

Dentre os onze elementos químicos quantificados pelo ICP-MS, o Mg foi o que apresentou maior concentração em ambos os sistemas de produção. Além disso, nas amostras cultivadas organicamente, é possível observar uma maior concentração de Cu, Mn, Cs, Fe e Si, enquanto que as amostras cultivadas convencionalmente apresentaram maior concentração de Al, Rb, Sr, Ba e Zn.

Na Tabela (7.3) são apresentados os melhores resultados, que foram obtidos utilizando o SVM combinado com $\chi^2$, alcançando 93,00% de acurácia usando somente sete dos componentes químicos obtidos (Cu, Cs, Zn, Al, Mn, Rb e Sr) e classificando corretamente 96,73% das amostras cultivadas no sistema orgânico. Na combinação do algoritmo SVM com o seletor *F-score*, foram necessárias dez variáveis para obter a mesma acurácia

de 93,00%, porém classificou corretamente 100,0% das amostras cultivadas organicamente.

**Tabela 7.3:** *Medidas de desempenho detalhadas com os melhores resultados, modelo calculado - SVM.*

| Subconjunto de variáveis | #F10 | #X7 |
|---|---|---|
| acurácia (%) | 93,00 | 93,00 |
| Sensitividade (%) | 85,71 | 89,12 |
| Especificidade (%) | 100,0 | 96,73 |

### 7.2.5 Conclusão

Na classificação das amostras de laranjas, usamos algoritmos combinados de seleção de variáveis (*F-score* e $\chi^2$) e algoritmos de classificação (SVM e MLP) em 57 amostras, utilizando os componentes químicos obtidos pelo ICP-MS. Onze elementos químicos foram determinados e foram calculadas algumas estatísticas, como média e desvio padrão, sendo que o Mg apresentou a maior concentração em ambos os sistemas de produção. Além disso, nas amostras cultivadas organicamente, foi possível observar uma maior concentração de Cu, Mn, Cs, Fe e Si, enquanto que nas amostras cultivadas convencionalmente foram encontradas maior concentração de Al, Rb, Sr, Ba e Zn. Os testes foram realizados e os melhores resultados foram obtidos com SVM combinado com $\chi^2$, apresentando uma acurácia de 93,00%, classificando corretamente 96,73% das amostras cultivadas organicamente, utilizando apenas sete variáveis. O SVM combinado com *F-score* classificou 100,0% das amostras organicas utilizando 10 variáveis e alcançou os mesmos 93,00% de acurácia.

## 7.3 Reconhecimento geográfico de vinhos Syrah produzidos na Argentina e Chile

### 7.3.1 Introdução

Vinhos são bebidas obtidas a partir da fermentação alcoólica das uvas. Em 2014, a Argentina foi considerada a quinta maior produtora de vinhos do mundo e o Chile ocupou a nona posição na classificação realizado pelo Wine Institute [65]. Entre as variedades de uvas *Vitis Vinífera*, Syrah é uma das mais utilizadas para a produção de vinhos [65]. Vinhos dessa variedade são amplamente consumidos, conforme apresenta o Instituto Nacional de Viticultura da Argentina [34], representando 5,8% de todos os vinhos produzidos no país. No Chile, a produção desses vinhos é maior, atingindo 8,1%

do total produzido em 2015, segundo o Ministério da Agricultura do Governo do Chile [135].

Durante as últimas décadas, o interesse em classificar os vinhos com base em sua variedade de uvas e origem geográfica aumentou. O solo, as condições climáticas, o tipo de colheita e as condições de produção contribuem para as características que tornam um vinho único. Este conceito está ligado ao chamado "Estatuto Geográfico Protegido" definido na legislação da União Europeia para vinhos e outros produtos alimentícios, que está expandindo gradualmente a nível internacional. A ideia é garantir que apenas vinhos originados em determinada região sejam permitidos no mercado, protegendo a reputação da região e assegurando um padrão de qualidade. Juntamente com este conceito, surgiu a necessidade de classificar os vinhos de forma objetiva, a fim de realizar controles de qualidade e evitar fraudes.

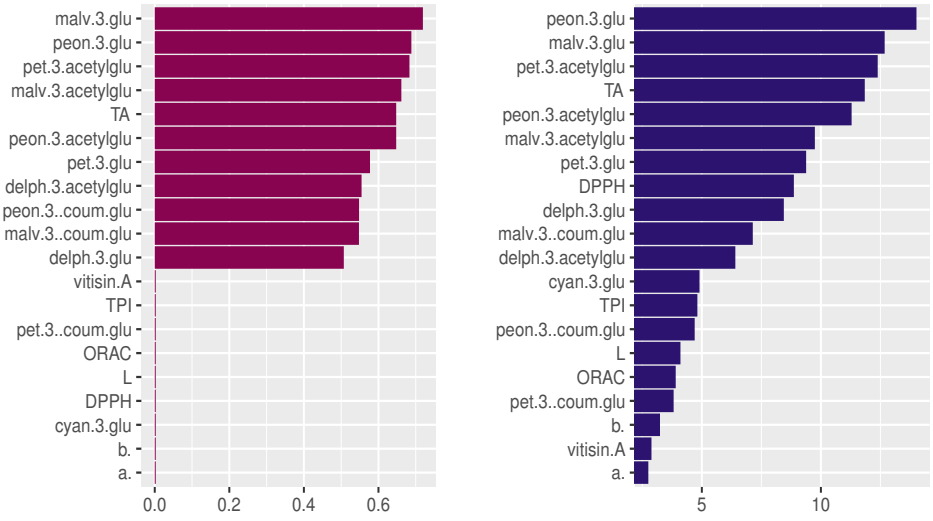
Esse estudo teve como objetivo encontrar um modelo capaz de classificar vinhos produzidos das uvas Syrah originários da Argentina, na região de Mendoza, e do Chile, na região do Vale Central, utilizando seletores de variáveis em suas componentes químicas, combinadas com algoritmos de classificação Support Vector Machines (SVM) e dois tipos de redes neurais, Multilayer Perceptron (MLP) e Extreme Learning Machine (ELM).

### 7.3.2 Conjunto de dados analisados

Foram analisadas 37 amostras neste estudo, sendo vinte e seis amostras oriundas da Argentina (Mendoza) e onze amostras do Chile (Vale Central). Foram obtidas 20 variáveis dessas amostras. As variáveis descrevem componentes como a cor do vinho (L, a., B.), Polifenóis totais (TPI), antocianinas totais (TA), atividade antioxidante pela capacidade de absorção de radicais livres de oxigênio (ORAC) e atividade eliminadora de radicais livres (DPPH) e antocianinas (cian-3-glu, delph-3-acetilglu, delph-3-glu, malv-3-(coum) glu, malv-3-acetilglu, malv-3-glu, peon-3- (coum) glu , peon-3-acetilglu, peon-3-glu, pet-3- (coum) glu, pet-3-acetilglu, pet-3-glu e vitisina A). As análises foram realizadas conforme descrito por Llobodanin et al. [93].

### 7.3.3 Metodologia

Foram obtidas 20 variáveis, utilizando os procedimentos quimiométricos, conforme descrito na Seção (7.3.2). Dois seletores de variáveis, $\chi^2$ Fig. (7.1a) e *random forest importance* (RFI) Fig. (7.1b), foram combinados com algoritmos de classificação, Support Vector Machines (SVM) e dois tipos de redes neural artificial, Multilayer Perceptron (MLP) e Extreme Learning Machine (ELM).




(a) (b)

**Fig 7.1:** *Gráficos dos rankings utilizando: (a)* $\chi^2$ *e (b) RFI.*

### 7.3.4 Resultados

Cada método seletor de variável tem uma abordagem diferente, pois cria diferentes conjuntos com as mais importantes variáveis (elementos químicos), ver Tabela (7.4) .

| Classificação | $\chi^2$ | RFI |
|---|---|---|
| 1º | malv-3-glu | peon-3-glu |
| 2º | peon-3-glu | malv-3-glu |
| 3º | pet-3-acetylglu | pet-3-acetylglu |
| 4º | malv-3-acetylglu | TA |
| 5º | TA | peon-3-acetylglu |
| 6º | peon-3-acetylglu | malv-3-acetylglu |
| 7º | pet-3-glu | pet-3-glu |
| 8º | delph-3-acetylglu | DPPH |
| 9º | peon-3-(coum)glu | delph-3-glu |
| 10º | malv-3-(coum)glu | malv-3-(coum)glu |
| 11º | delph-3-glu | delph-3-acetylglu |
| 12º | | cyan-3-glu |
| 13º | | TPI |
| 14º | | peon-3-(coum)glu |
| 15º | | L |
| 16º | | ORAC |
| 17º | | pet-3-(coum)glu |
| 18º | | b. |
| 19º | | vitisin A |
| 20º | | a. |

**Tabela 7.4:** *Classificação de importância das variáveis segundo* $\chi^2$ *e RFI.*

O número de variáveis selecionadas para a construção do modelo de classificação está diretamente relacionado ao desempenho dos classificadores, como podemos ver na Fig. (7.2).

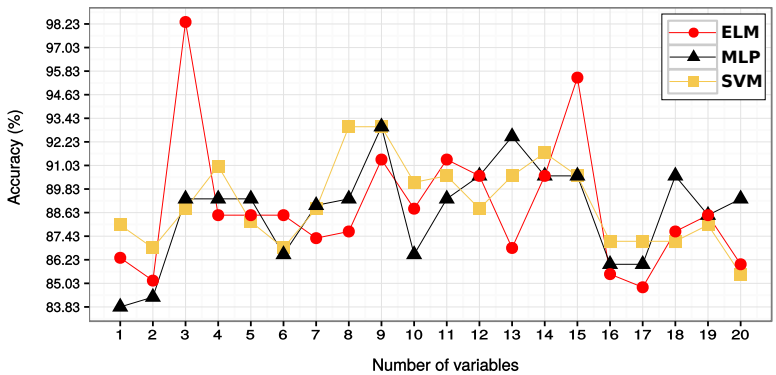



**Fig 7.2:** *Acurácia do SVM, MLP e ELM utilizando o seletor RFI.*

O melhor modelo foi obtido pela combinação das variáveis peon-3-glu, malv-

3-glu e pet-3-acetylglu, selecionadas pelo seletor RFI, atingindo 98.33% de acurácia com o ELM, superando o SVM e MLP, ver Tabela (7.5). Esse estudo também mostrou a importância das antocianinas para classificar vinhos Syrah de acordo com sua região geográfica.

**Tabela 7.5:** *Melhores resultados (em %) obtidos com SVM, MLP e ELM associado com seleção de variáveis.*

| | $\chi^2$ | | RFI | | Todas as variáveis | |
|---|---|---|---|---|---|---|
| | Variáveis | acurácia (%) | Variáveis | acurácia (%) | Variáveis | acurácia (%) |
| SVM | 11 | 89,67 | 7 | 93,00 | 20 | 85,50 |
| MLP | 10 | 93,50 | 9 | 93,00 | 20 | 91,83 |
| ELM | 10 | 93,50 | 3 | 98,33 | 20 | 89,33 |

### 7.3.5 Conclusão

Os vinhos Syrah foram classificados de acordo com a região de produção, dois métodos de seleção de variáveis foram aplicados a três classificadores diferentes. Usando apenas três variáveis o algoritmo ELM classificou os vinhos com uma acurácia de 98,33%. Adicionalmente, foi possível inferir quais variáveis possuem maior capacidade discriminatória entre as regiões geográficas de origem dos Vinhos Syrah. O modelo final conseguiu atribuir a origem geográfica a uma nova amostra de vinho usando apenas três antocianinas.

## 7.4 Impressão digital e técnicas de aprendizado de máquina para avaliar a autenticidade de vegetais orgânicos e convencionais

### 7.4.1 Introdução

Pesquisadores ao avaliarem a qualidade dos alimentos, se mostram preocupados com o uso de pesticidas, antibióticos e hormônios utilizados na agricultura, juntamente com modificação genética e aditivos no processamento de alimentos [6]. O número de pessoas que preferem alimentos orgânicos está aumentando, e essa demanda forte e crescente levou a um aumento no número de agricultores comprometidos com a agricultura orgânica [11].

A intensificação do setor orgânico, impulsionada pela demanda crescente e pelos altos preços dos produtos orgânicos, poderia estimular a ocorrência de fraudes [82]. Por exemplo, um vegetal cultivado convencionalmente poderia ser erroneamente rotulado

como orgânico, visto que não há diferenças visuais entre esses produtos. Por causa disso, a autenticação de alimentos é de interesse para muitas comunidades científicas, órgãos reguladores e, principalmente, consumidores. Portanto, há uma necessidade crescente de pesquisas e desenvolvimento de técnicas analíticas de detecção rápidas [123]. Os métodos de autenticação de alimentos devem ser cientificamente comprovados, precisos e confiáveis para que produtos genuínos possam ser protegidos e diferenciados de alterações ilegais [18, 33].

No Brasil, a agricultura orgânica que não possui certificação pode fazer parte de uma Organização de Controle Social (SCO), definida como:

> "grupo, associação, cooperativa ou consórcio a que está vinculado o agricultor familiar em venda direta, previamente cadastrado no Ministério da Agricultura, Pecuária e Abastecimento, com processo organizado de geração de credibilidade a partir da interação de pessoas ou organizações, sustentado na participação, comprometimento, transparência e confiança, reconhecido pela sociedade" (Brasil, 2007, Art. 2°, § VIII) [104].

Neste tipo de negócio, o consumidor negocia diretamente com o produtor e tem que confiar na qualidade do produto. Uma situação como essa mostra a importância do processo de autenticação para garantir a origem orgânica dos produtos.

Com isso, nosso objetivo é utilizar algoritmos de aprendizado de máquina para classificar amostras de vegetais produzidos nos sistemas orgânicos e convencionais com base em sua composição química.

### 7.4.2 Conjunto de dados analisados

Foram utilizadas 158 amostras de alface (*Lactuca sativa*) de oito locais cultivados convencionalmente e oito locais cultivados organicamente (perfazendo um total de 80 e 78 amostras, respectivamente); 106 amostras de cebolas (*Allium cepa Hysam*) de oito locais com cultivo convencional e três cultivados organicamente (perfazendo um total de 76 e 30 amostras, respectivamente); 56 amostras de pimentão (*Capsicum annuum*) de cinco locais cultivados convencionalmente e um organicamente (totalizando 46 e 10 amostras, respectivamente); e mais 29 amostras de tomate (*Solanum lycopersicum*), sendo 19 amostras cultivadas convencionalmente e 10 amostras cultivadas organicamente. Todas essas amostras foram originadas de fazendas do estado do Pernambuco para evitar diferenças devido à variação geográfica. As amostras consideradas "orgânicas" foram obtidas de fazendas orgânicas não certificadas (SCO). Foram também utilizadas dez amostras de alface orgânica e cinco de pimentão todas com certificação orgânica em Recife (capital do estado de Pernambuco).

### 7.4.3 Metodologia

A quantificação dos elementos químicos (Al, As, B, Ba, Ca, Cd, Co, Cr, Cu, Fe, Hg, K, Mg, Mn, Mo, Na, Ni, P, Pb, S, Se, Si, Ti, V , Zn) das amostras foi obtidas por um processo de Espectrometria de Emissão óptica por Plasma Acoplado Indutivamente (ICP-OES) usando o sistema convencional de introdução de amostras.

O método *leave-one-out* (LOOCV) [75] é outra versão da validação cruzada $K-fold$, em que $K$ é igual ao número de amostras, o processo consiste em usar $K-1$ amostras para treinamento e o restante é usado para o teste. Este processo é realizado até que todas as amostras sejam testadas [17].

Primeiramente, os dados foram balanceados para fornecer quantidades proporcionais de amostras. As amostras foram normalizadas entre [0-1], uma vez que houve uma variação muito grande entre as quantificações obtidas. A próxima etapa foi usar a seleção de variáveis e a aplicação dos algoritmos de classificação (SVM, MLP e RF).

### 7.4.4 Resultados

Em relação às amostras de tomate, o SVM combinado com o seletor $\chi^2$ obteve acurácia de 97,00%, usando apenas dois elementos (Si e Ca); amostras de pimentão atingiram 100,0% com seis elementos selecionados pelo $\chi^2$ (Si, Ca, B, Mg, P e Na). Amostras de cebola necessitaram de sete elementos (Mn, Ca, Fe, Ni, Na, Zn e Cu), selecionadas pelo $\chi^2$, e também atingiram 100,0% de acurácia, enquanto amostras de alface atingiram uma taxa de acurácia de 92,00% com seis elementos selecionados pelo $\chi^2$ (Si, B, S, Na, Cu e Ba).

A Fig. (7.3) apresenta o comportamento das amostras de cebola usando apenas dois elementos selecionados pelo $\chi^2$ (Ca e Mn) e obteve uma acurácia de 98,0%, usando o SVM com o *kernel* RBF.

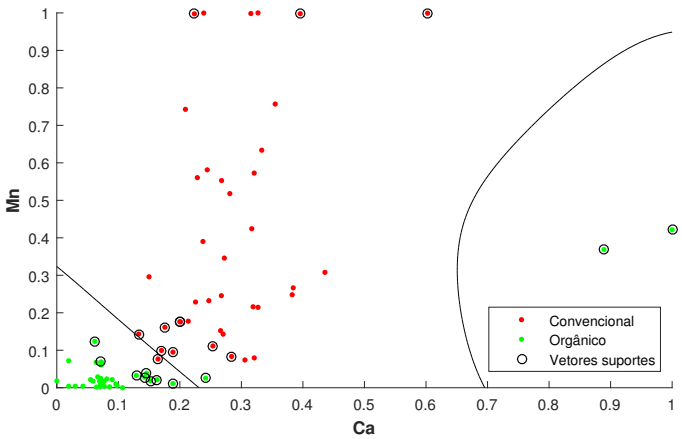



**Fig 7.3:** *Comportamento do classificador SVM em relação às variáveis Ca e Mn, obtidas das amostras de cebola.*

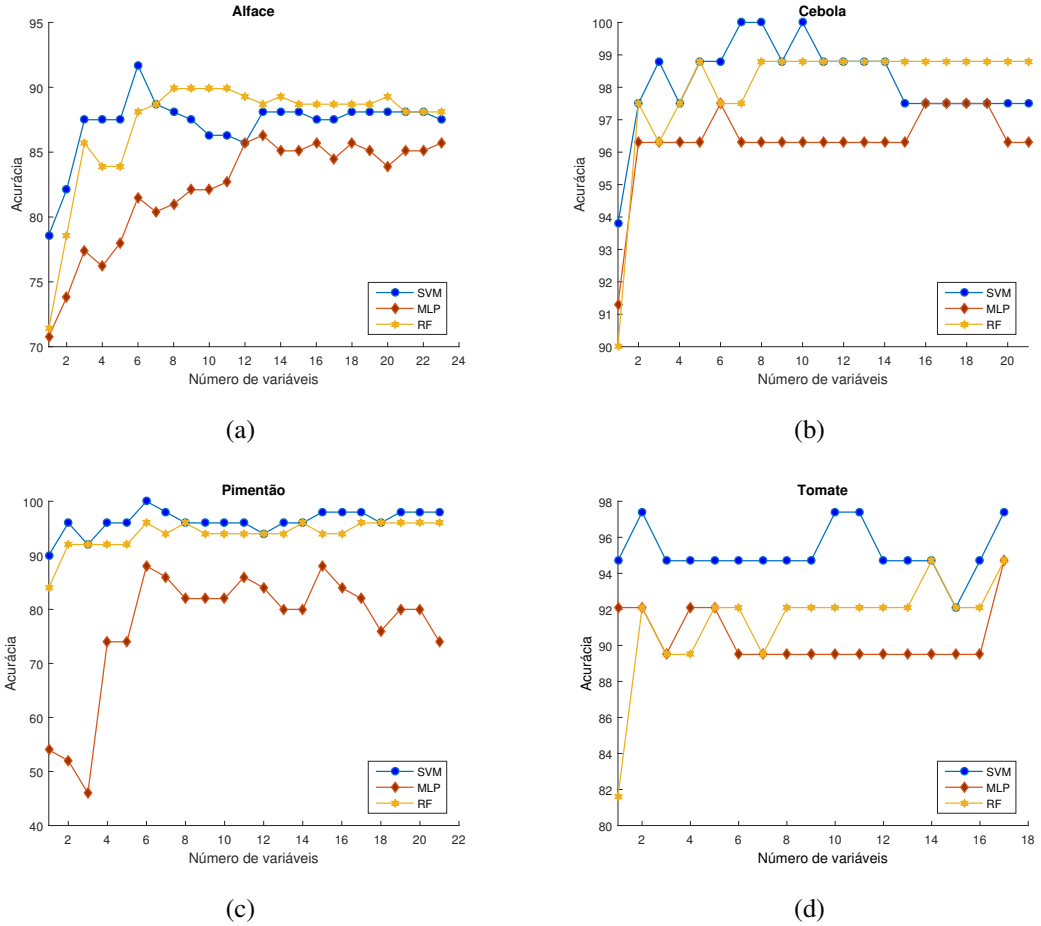



**Fig 7.4:** *Gráfico dos classificadores (SVM, MLP e RF) com variáveis selecionadas pelo $\chi^2$ (a) amostras de alface, (b) amostras de cebola, (c) amostras de pimentão e (d) amostras de tomate.*

Fig. (7.4) mostra o comportamento dos classificadores em relação aos números de variáveis e as acurácias obtidas. SVM combinado com o $\chi^2$ superou MLP e RF em todas as bases de dados, atingindo acurácia de 97,50% nas amostras de tomate, usando apenas duas variáveis (Si e Ca). As amostras de pimentão obteve acurácia de 100,0% usando seis variáveis (Si, Zn, Ca, Ni, Na e Cu), as amostras de cebola obtiveram 100,0% de acurácia, utilizando sete variáveis (Mn, Ca, Fe, Ni, Na, Zn e Cu), e para as amostras de alface, a acurácia foi de 91,70% usando seis variáveis.

Tabela (7.6) apresenta as medidas de acurácia, sensitividade e especificidade. O SVM supera o MLP e RF nas amostras de tomate, pimentão, cebola e alface alcançando uma acurácia de (97,50%, 100,0%, 100,0% e 91,70%), respectivamente. Mais de 90,00% das amostras orgânicas de tomate, pimentão, cebola e alface foram corretamente classificadas (especificidade) (94,70%, 100,0%, 100,0% e 90,90%, respectivamente).

**Tabela 7.6:** *Medidas de desempenho detalhadas com os melhores resultados (em %).*

| Bases de dados | Tomate | Pimentão | Cebola | Alface |
|---|---|---|---|---|
| Subconjunto de variáveis | #2 ($\chi^2$) | #6 ($\chi^2$) | #7 $\chi^2$) | #6 ($\chi^2$) |
| Classificador | SVM | SVM | SVM | RF |
| Acurácia (%) | 97,50 | 100,0 | 100,0 | 91,70 |
| Sensitividade (%) | 100,0 | 100,0 | 100,0 | 92,50 |
| Especificidade (%) | 94,70 | 100,0 | 100,0 | 90,90 |

Os resultados obtidos mostram que a técnica de mineração de dados, que combina algoritmo de classificação e seleção de variáveis, é muito eficaz para discriminar amostras que foram cultivadas nos sistemas orgânicos e convencionais.

### 7.4.5 Conclusão

As técnicas utilizadas neste estudo foram eficientes (níveis de acurácia acima de 90,00%) na classificação de tomate, pimentão cebola e alface cultivadas em sistemas orgânico e convencional. O SVM superou o MLP e RF para todos os quatro grupos de amostras (tomate, pimentão, cebola e alface). O número reduzido de elementos necessários para classificar os diferentes tipos como culturas convencionais ou orgânicas é um ponto positivo importante, pois evidencia o baixo custo operacional do processo de autenticação desses vegetais.

Considerando o cenário de produção de hortaliças no Brasil (onde a comercialização de produtos orgânicos nem sempre está estritamente vinculada à certificação de selos), nosso estudo oferece uma alternativa viável aos órgãos fiscalizadores, pois conseguimos classificar esses quatro vegetais importantes comumente consumidos por humanos, combinando a quantificação da composição química com técnicas de aprendizado de máquina.

## 7.5 Classificação em bases de dados médicos, usando um novo algoritmo de seleção de variáveis, combinado com máquinas de vetores de suporte duplo

### 7.5.1 Introdução

O diagnóstico e tratamento precoce são as estratégias mais importantes para prevenir mortes por várias doenças. A esse respeito, técnicas de mineração de dados e de

aprendizado de máquina têm sido ferramentas úteis para ajudar a minimizar erros e fornecer informações úteis para o diagnóstico. Com o propósito de contribuir na construção de ferramentas que possam auxiliar na seleção de variáveis que mais influenciam nesses diagnósticos, neste capítulo desenvolvemos um novo algoritmo de seleção de variáveis, baseado no TBSVM.

No algoritmo TBSVM é implementado o princípio de minimização do risco estrutural, conforme apresentado no Capítulo 4. Além disso, com o intuito de acelerar o tempo de treinamento, o algoritmo *successive overrelaxation* (SOR) proposto por Mangasarian e Musicant [99] é utilizado na resolução dos PPQs do TBSVM. Este projeto culminou em um artigo que está submetido.

### 7.5.2 Seletor de variáveis baseado no twin bounded support vector machine - FSTBSVM

No aprendizado de máquina, o problema de seleção de variáveis é bem conhecido e há algumas maneiras de realizar esse procedimento [182, 12]. Uma delas é a técnica dependente de algoritmo, como no SVM, com base na eliminação recursiva de variáveis, em inglês: *recursive feature elimination* (RFE) [53], ou com técnicas independentes de algoritmos, tal como o *F-score* [26]. Esse último consiste em técnicas de classificação de variáveis particularmente atraente [53]. Na literatura encontramos poucos estudos sobre seleção de variáveis combinados com TWSVM ou algumas de suas variantes [182, 51, 154, 7, 97].

Dados um vetor de treinamento $x_k$, $k = 1,\ldots,m$, $A \in \mathbb{R}^{n_+ \times m}$ a classe rotulada por $+1$ e $B \in \mathbb{R}^{n_- \times m}$ a classe rotulada por $-1$, em que $n_+$ e $n_-$ são os números de amostras positivas e negativas, respectivamente. O processo consiste em realizar o ranking para cada classe separadamente por meio das pontuações obtidas para cada variável e depois é calculado a média entre essas pontuações. Dessa maneira, a pontuação para a $i$-ésima amostra positiva ou negativa são definidas pelas Eqs. (7-1) e (7-2), respectivamente.

$$R_{positivas}(i) = \frac{(\bar{x}_i^{(+)} - \bar{x}_i)^2}{\frac{1}{n_+ - 1}\sum_{k=1}^{n_+}\left(x_{k,i}^{(+)} - \bar{x}_i^{(+)}\right)^2} \tag{7-1}$$

e

$$R_{negativas}(i) = \frac{(\bar{x}_i^{(-)} - \bar{x}_i)^2}{\frac{1}{n_- - 1}\sum_{k=1}^{n_-}\left(x_{k,i}^{(-)} - \bar{x}_i^{(-)}\right)^2}, \tag{7-2}$$

em que $\overline{x}_i$ é a média da $i$-ésima variável em relação a todo o conjunto de dados, $\overline{x}_i^{(+)}, \overline{x}_i^{(-)}$ são as médias das amostras positivas e negativas, respectivamente. $x_{k,i}^{(+)}$ e $x_{k,i}^{(-)}$ são as $i$-ésima variável da $k$-ésima amostra positiva e negativa, respectivamente.

O ranking das variáveis (RV) é obtido pela Eq. (7-3), onde o maior valor da pontuação obtida, indica a maior importância da variável,

$$RV(i) = \frac{R_{positivas}(i) + R_{negativas}(i)}{2}. \tag{7-3}$$

O procedimento para calcular o ranking das variáveis para ser utilizado no TBSVM é apresentado no Algoritmo (7.5.1) como segue:

**Algoritmo 7.5.1** *Procedimento para o FSTBSVM*

1. *Calcule as pontuações das variáveis em relação as amostras positivas e negativas utilizando as Eqs. (7-1) e (7-2), respectivamente;*
2. *Calcule o ranking das variáveis usando a Eq. (7-3);*
3. *Treine o classificador (TBSVM) usando as variáveis com o maior ranking e calcule sua acurácia;*
4. *Depois disso, calcule a acurácia adicionando mais uma variável com a segunda maior pontuação, repita esse processo até que todas as variáveis sejam usadas.*

### 7.5.3 Conjuntos de dados analisados

Para validar nosso estudo, utilizamos oito conjuntos de dados, comumente usados por pesquisadores que desenvolveram métodos de aprendizado de máquina para classificação de dados médicos, obtidos no repositório da *UCI Irvine* [87]. Na Tabela (7.7) são apresentadas as dimensões de cada bases de dados.

**Tabela 7.7:** *Descrição das bases de dados utilizadas.*

| Bases de dados | Número de variáveis | Amostras positivas | Amostras negativas |
|---|---|---|---|
| *BCWO* [1] | 9 | 241 | 458 |
| *Diabetic Retinopathy* | 19 | 611 | 540 |
| *Pima Indians Diabetes* | 8 | 500 | 268 |
| *Immunotherapy* | 8 | 71 | 19 |
| *Heart-Statlog* | 13 | 150 | 120 |
| *Hepatitis* | 19 | 32 | 123 |
| *WDBC* [2] | 30 | 357 | 212 |
| *Fertility* | 10 | 88 | 12 |

### 7.5.4 Metodologia

A Fig. (7.5) apresenta um esquema metodológico utilizado, onde primeiramente os dados são normalizados entre $[0, 1]$, logo após utilizamos o FSTBSVM para o ranking das variáveis criando subconjuntos, sendo o primeiro subconjunto contendo a variável com a maior pontuação, o segundo subconjunto contendo as duas variáveis com as maiores pontuações e assim por diante até contemplar todas as variáveis.

Com os subconjuntos gerados, utilizamos o método de busca em grade para obter os melhores parâmetros para cada subconjunto de amostras, durante o processo de treinamento do algoritmo. O próximo passo é identificar os melhores parâmetros e o melhor subconjunto de variáveis e assim calcular a acurácia, sensitividade e especificidade para cada base de dados.

Como o TBSVM gera dois hiperplanos não paralelos e possui desempenho superior ao TWSVM, o FSTBSVM proposto, além de utilizar essa vantagem, também realiza o cálculo das pontuações de cada variável individualmente em relação às amostras positivas e negativas, desse modo, cada hiperplano determina seu ranking e, em seguida, a média das duas pontuações são calculadas, fornecendo uma pontuação final a ser usada no TBSVM.

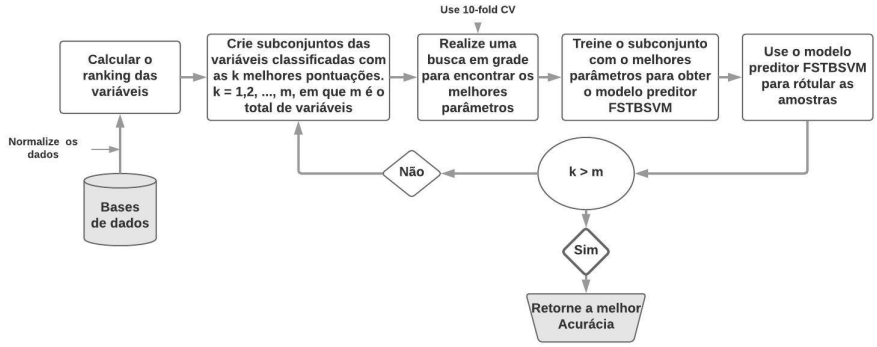



**Fig 7.5:** *Fluxograma do nosso FSTBSVM usando a busca em grade para encontrar o modelo otimizado.*

### 7.5.5 Resultados

Os experimentos mostraram que o desempenho do nosso método de seleção de variáveis proposto é muito eficiente. A robustez do FSTBSVM é examinada usando as medidas de acurácia, sensitividade e especificidade. O FSTBSVM proposto é uma técnica muito interessante, pois é capaz de produzir bons resultados com menos variáveis do que com os conjuntos de dados originais.

[1]BCWO *Breast Cancer Wisconsin (Original)*.
[2]WDBC *Breast Cancer Wisconsin (Diagnostic)*.

A Tabela (7.8) apresenta os resultados dos experimentos em oito bases de dados comparando o FSTBSVM, TBSVM e TWSVM. A acurácia é obtida da média das acurácias dos resultados das dez vezes que os testes foram executados, seguido de mais ou menos o desvio padrão. São apresentados também os parâmetros utilizados para obter as melhores acurácias. O tempo apresentado é a média dos tempos obtidos, em segundos, para executar os dez experimentos, incluindo tempo de treino e teste.

Em alguns casos, o melhor desempenho foi obtido usando somente a metade das variáveis, conforme apresentado em relação à base de dados *Fertility*. Embora a base de dados *Pima Indians Diabetes* tenha alcançado 78,51% usando todas as variáveis, apresentou um excelente desempenho com metade das variáveis, atingindo 77,21% de acurácia. No conjunto de dados *Immunotherapy*, apenas uma variável foi necessária para obter a melhor precisão de 87,89%, em comparação com as oito variáveis que obtiveram 83,41% de acurácia. Devemos também destacar o conjunto de dados *Diabetic Retinopathy* que precisou de menos da metade de suas variáveis para superar a acurácia obtida, quando todas as variáveis foram utilizadas, ou seja, 73,67%. O WDBC teve desempenho comparável para ambos os algoritmos, alcançando 98,24%, mas apenas 21 variáveis foram necessárias ao usar o FSTBSVM.

**Tabela 7.8:** *Comparação de desempenho de conjuntos de dados com algoritmos usando o kernel RBF.*

| Bases de dados | FSTBSVM | | TBSVM | | TWSVM | |
|---|---|---|---|---|---|---|
| | Número de variáveis | Acurácia ± D. P.<br>$\gamma, c_1, c_2, c_3, c_4$<br>Tempo (s) | Número de variáveis | Acurácia ± D. P.<br>$\gamma, c_1, c_2, c_3, c_4$<br>Tempo (s) | Número de variáveis | Acurácia ± D. P.<br>$\gamma, c_1, c_2$<br>Tempo (s) |
| *BCWO* | 8 | 97,57±1,51<br>$2^5, 2^{-4}, 2^{-4}, 2^6, 2^{-3}$<br>0,21 | 9 | 97,42±1,62<br>$2^5, 2^{-4}, 2^{-4}, 2^6, 2^{-3}$<br>0,22 | 9 | 92,71±5,10<br>$2^{-6}, 2^0, 2^{-8}$<br>23,20 |
| *Diabetic Retinopathy* | 9 | 73,76±3,97<br>$2^{-6}, 2^{-6}, 2^{-2}, 2^{-6}, 2^{-6}$<br>0,55 | 19 | 70,80±3,81<br>$2^{-2}, 2^0, 2^{-2}, 2^{-6}, 2^{-2}$<br>0,69 | 19 | 59,57±3,78<br>$2^{-6}, 2^{-6}, 2^{-2}$<br>69,51 |
| *Pima Indians Diabetes* | 4 | 77,21±3,31<br>$2^{-6}, 2^{-2}, 2^{-6}, 2^0, 2^5$<br>0,38 | 8 | 78,51±5,37<br>$2^0, 2^0, 2^0, 2^{-2}, 2^{-2}$<br>0,30 | 8 | 68,75±2,71<br>$2^{-6}, 2^0, 2^{-4}$<br>2,71 |
| *Immunotherapy* | 1 | 87,89±8,22<br>$2^2, 2^2, 2^2, 2^5, 2^2$<br>0,04 | 8 | 83,41±13,97<br>$2^2, 2^2, 2^0, 2^5, 2^5$<br>0,03 | 8 | 78,93±1,98<br>$2^{-6}, 2^{-8}, 2^4$<br>0,41 |
| *Heart-Statlog* | 8 | 85,92±6,24<br>$2^2, 2^{-6}, 2^0, 2^5, 2^2$<br>0,02 | 13 | 84,44±5,74<br>$2^2, 2^{-6}, 2^0, 2^5, 2^2$<br>0,02 | 13 | 69,62±7,81<br>$2^{-6}, 2^{-2}, 2^{-4}$<br>3,46 |
| *Hepatitis* | 13 | 89,15±6,60<br>$2^2, 2^2, 2^{-2}, 2^{-6}, 2^{-6}$<br>0,02 | 19 | 86,57±10,13<br>$2^0, 2^{-2}, 2^{-2}, 2^{-2}, 2^{-6}$<br>0,008 | 19 | 81,30±3,39<br>$2^{-6}, 2^0, 2^{-8}$<br>1,12 |
| *WDBC* | 21 | 98,24±1,65<br>$2^0, 2^2, 2^{-2}, 2^0, 2^0$<br>0,65 | 30 | 98,24±1,85<br>$2^2, 2^2, 2^{-2}, 2^{-2}, 2^{-2}$<br>0,09 | 30 | 74,34±1,75<br>$2^{-6}, 2^{-2}, 2^{-8}$<br>16,15 |
| *Fertility* | 5 | 91,89±7,92<br>$2^2, 2^{-6}, 2^5, 2^5, 2^5$<br>0,02 | 10 | 88,98±3,19<br>$2^{-2}, 2^{-6}, 2^{-2}, 2^5, 2^2$<br>0,005 | 10 | 88,14±5,29<br>$2^{-6}, 2^{-8}, 2^0$<br>0,74 |
| Acurácia Média | | 87,70 ± 4,93 | | 86,04 ± 5,71 | | 76,67 ± 3,98 |

**Tabela 7.9:** *Valores obtidos da acurácia, sensitividade e especificidade utilizando o FSTBSVM com kernel RBF.*

| Bases de dados | Acurácia (em %) | Sensitividade (em %) | Especificidade (em %) |
|---|---|---|---|
| *BCWO* | 97,57 | 91,30 | 95,74 |
| *Diabetic Retinopathy* | 73,76 | 71,70 | 82,25 |
| *Pima Indians Diabetes* | 77,21 | 87,21 | 71,06 |
| *Immunotherapy* | 87,89 | 100,0 | 87,50 |
| *Heart-Statlog* | 85,93 | 72,73 | 93,75 |
| *Hepatitis* | 89,15 | 78,57 | 100,0 |
| *WDBC* | 98,24 | 100,0 | 97,30 |
| *Fertility* | 91,89 | 100,0 | 88,89 |

A Tabela (7.9) apresenta a acurácia, sensitividade e especificidade obtidas, com os melhores parâmetros para cada base de dados. Podemos destacar os conjuntos de dados *Immunotherapy* e *WDBC*, que obtiveram 100% de sensitividade e especificidade, utilizando apenas 12,50% e 70,00% das variáveis nas respectivas bases de dados. A base de dados *Fertility* atingiu 100% de sensitividade e 88,89% de especificidade usando somente 50,00% das variáveis, enquanto que a base *Hepatitis* obteve 78,57% de sensitividade e 100% de especificidade, com somente 68,42% das variáveis da base de dados original. A base *BCWO* obteve sensitividade e especificidade (>91,00%) utilizando somente 88,89% das variáveis totais.

Realizamos uma comparação com outros estudos que utilizaram o conjunto de dados *BCWO*. Os resultados obtidos pelos vários métodos são apresentados na Tabela (7.10) e o FSTBSVM apresenta acurácia comparável com outros métodos encontrados na literatura.

**Tabela 7.10:** *Comparativo entre as acurácias obtidas, utilizando o conjunto de dados BCWO.*

| Autor (Ano) | Método | Partição | Acurácia (%) | Referências |
|---|---|---|---|---|
| Quinlan (1996) | C4.5 | (10-CV) | 94,74 | [130] |
| Hamilton et al. (1996) | RIAC | (leave-one-out) | 94,99 | [54] |
| Bennett and Blue (1997) | SVM | (5-CV) | 97,20 | [8] |
| Pena-Reyes and Sipper (1999) | Fuzzy-GA1 | (treino:75% teste:25%) | 97,36 | [114] |
| Goodman et al. (2002) | Optimized-LVQ | (10-CV) | 96,70 | [48] |
| Goodman et al. (2002) | Big-LVQ | (10-CV) | 96,80 | [48] |
| Goodman et al. (2002) | AIRS | (10-CV) | 97,20 | [48] |
| Abonyi and Szeifert (2003) | Supervised fuzzy clustering | (10-CV) | 95,57 | [1] |
| Polat and Güneş (2006) | LS-SVM | (10-CV) | 98,53 | [125] |
| Guo and Nandi (2006) | MF-GP/MDC | (10-CV) | 97,47 | [50] |
| Akay (2009) | F-score + SVM | (treino:80% teste:20%) | 99,51 | [2] |
| Chen et al. (2011) | RS_SVM | (treino:80% teste:20%) | 96,87 | [19] |
| Li et al. (2013) | QKCLDA | (7-CV) | 97,20 | [86] |
| Zheng et al. (2014) | K-SVM | (10-CV) | 97,40 | [193] |
| Peng et al. (2016) | Immune learning | (10-CV) | 98,00 | [115] |
| Nosso Estudo | FSTBSVM | (10-CV) | 97,57 | |

### 7.5.6 Conclusão

Propusemos um novo algoritmo de seleção de variáveis baseado no TBSVM. Nosso FSTBSVM realiza o ranking para cada classe de amostras e, em seguida, a média entre duas classificações é calculada e assim é aplicado no algoritmo TBSVM. Oito conjuntos de dados foram usados para validar nosso algoritmo. O FSTBSVM proposto

reduz significativamente o número de variáveis, obtendo ótimo desempenho no processo de classificação. Os experimentos mostraram que o desempenho do FSTBSVM proposto alcançou uma acurácia média superior a 87,00%, superando os outros métodos, TBSVM (86,04%) e TWSVM (76,67%). Esses resultados mostram que o método proposto é capaz de produzir bons resultados, utilizando menos variáveis.

## 7.6 Resultados comparativos entre as bases de dados de alimentos e vinhos, utilizando SVM, MLP, RF e FSTBSVM

Nesta seção iremos apresentar os resultados comparativos entre alguns dos métodos clássicos (SVM, MLP, RF) utilizados nos trabalhos citados nas seções anteriores, e compará-los com os com resultados obtidos pelo FSTBSVM quanto à classificação ao sistemas de cultivo, orgânico e convencional, nas amostras de laranja, tomate, pimentão, cebola e alface, e quanto à origem geográfica em relação as amostras de vinhos produzidos das uvas syrah cultivadas na Argentina e Chile.

Em relação as amostras de laranjas, a ordem das variáveis selecionadas pelo *F-score* foram: (“Si, Cu, Cs, Mn, Fe, Zn, Sr, Al, Ba, Rb, Mg”), enquanto o FSTBSVM ordenou do seguinte modo: ("Zn, Rb, Cs, Mg, Ba, Sr, Fe, Cu, Al, Si e Mn"). Os resultados obtidos, utilizando os classificadores SVM e *Multilayer perceptron* (MLP) combinados com o F-score e o FSTSBSM, são apresentados na Tabela (7.11). O conjunto #F1 é composto pela variável com o maior *score*, ou seja, a variável "Si"; o subconjunto #F2 é composto pelas variáveis "Si e Cu"; e assim por diante, até o subconjunto #F11 contemplando todas as variáveis. De modo análogo, é formado os subconjuntos das variáveis para serem utilizadas no FSTBSVM, sendo #F1* contendo a variável "Zn"até #F11* contendo todas as variáveis. Essa notação será utilizada também nas próximas tabelas.

**Tabela 7.11:** *Comparativo entre as acurácias (em %) obtidas pelo SVM, MLP e FSTBSVM nas amostras de laranjas.*

| Subconjuntos | SVM | MLP | Subconjuntos | FSTBSVM |
|---|---|---|---|---|
| #F01 | 68,33 | 68,66 | #F01* | 49,00 |
| #F02 | 82,33 | 83,00 | #F02* | 51,00 |
| #F03 | 80,33 | 81,67 | #F03* | 50,67 |
| #F04 | 80,00 | 81,00 | #F04* | 54,67 |
| #F05 | 82,33 | 75,67 | #F05* | 61,67 |
| #F06 | 82,33 | 76,00 | #F06* | 61,67 |
| #F07 | 85,67 | 82,33 | #F07* | 79,33 |
| #F08 | 84,33 | 77,67 | #F08* | 82,33 |
| #F09 | 82,33 | 75,67 | #F09* | 82,67 |
| #F10 | 93,00 | 80,67 | #F10* | 88,00 |
| #F11 | 91,67 | 75,67 | #F11* | 81,00 |

Podemos notar que embora o FSTBSVM não supere o SVM, ele obteve desempenho comparável ao MLP em relação as acurácias obtidas, porém foi necessário um conjunto contendo dez variáveis para atingir os 88.00% de acurácia.

**Tabela 7.12:** *Comparativo entre as acurácias (em %) obtidas pelo SVM, MLP, RF e FSTBSVM nas amostras de tomate.*

| Subconjuntos | SVM | MLP | RF | Subconjuntos | FSTBSVM |
|---|---|---|---|---|---|
| #F01 | 95 | 92 | 82 | #F01* | 57,50 |
| #F02 | 92 | 89 | 84 | #F02* | 74,17 |
| #F03 | 95 | 89 | 89 | #F03* | 69,17 |
| #F04 | 95 | 89 | 89 | #F04* | 63,33 |
| #F05 | 95 | 92 | 92 | #F05* | 70,83 |
| #F06 | 95 | 92 | 89 | #F06* | 81,67 |
| #F07 | 95 | 89 | 89 | #F07* | 94,17 |
| #F08 | 97 | 97 | 95 | #F08* | 85,00 |
| #F09 | 97 | 97 | 95 | #F09* | 75,00 |
| #F10 | 95 | 95 | 92 | #F10* | 73,33 |
| #F11 | 97 | 97 | 95 | #F11* | 67,50 |
| #F12 | 97 | 97 | 92 | #F12* | 67,50 |
| #F13 | 97 | 95 | 92 | #F13* | 72,50 |
| #F14 | 97 | 95 | 92 | #F14* | 70,83 |
| #F15 | 97 | 95 | 92 | #F15* | 62,50 |
| #F16 | 97 | 95 | 92 | #F16* | 62,50 |
| #F17 | 97 | 95 | 95 | #F17* | 65,83 |

Nas Tabelas (7.12)-(7.15) são apresentados os resultados em relação aos classificadores SVM, MLP, RF que utilizaram o *F-score* como seletor de variáveis e o FSTBSVM. Nas amostras de Pimentão o FSTBSVM se apresentou bem interessante (Tabela 7.13), pois atingiu uma precisão de 98,33% utilizando apenas seis variáveis, enquanto o SVM para alcançar 98,00% utilizou dezenove variáveis. Já o RF atingiu 100,00% de acurácia utilizando doze variáveis. Nas amostras de cebola, embora o FSTBSVM não supere os outros algoritmos em acurácia (Tabela 7.14), os resultados obtidos são muito satisfatórios, pois alcançam 93,75% utilizando um conjunto com onze variáveis. Na Tabela (7.15) são apresentados os resultados obtidos em relação as amostras de alface. Podemos notar que os desempenhos obtidos pelos algoritmos são bem similares.

**Tabela 7.13:** *Comparativo entre as acurácias (em %) obtidas pelo SVM, MLP, RF e FSTBSVM nas amostras de pimentão.*

| Subconjuntos | SVM | MLP | RF | Subconjuntos | FSTBSVM |
|---|---|---|---|---|---|
| #F01 | 68 | 62 | 54 | #F01* | 50,00 |
| #F02 | 80 | 76 | 56 | #F02* | 54,50 |
| #F03 | 80 | 74 | 58 | #F03* | 62,67 |
| #F04 | 82 | 80 | 76 | #F04* | 64,67 |
| #F05 | 76 | 78 | 76 | #F05* | 72,33 |
| #F06 | 80 | 80 | 94 | #F06* | 98,33 |
| #F07 | 82 | 78 | 92 | #F07* | 82,00 |
| #F08 | 78 | 76 | 94 | #F08* | 68,17 |
| #F09 | 94 | 72 | 96 | #F09* | 68,50 |
| #F10 | 94 | 78 | 98 | #F10* | 58,33 |
| #F11 | 94 | 78 | 98 | #F11* | 58,33 |
| #F12 | 94 | 76 | 100 | #F12* | 52,33 |
| #F13 | 92 | 76 | 100 | #F13* | 52,33 |
| #F14 | 92 | 76 | 100 | #F14* | 54,00 |
| #F15 | 92 | 82 | 100 | #F15* | 54,00 |
| #F16 | 94 | 80 | 100 | #F16* | 50,00 |
| #F17 | 96 | 76 | 96 | #F17* | 50,00 |
| #F18 | 94 | 76 | 96 | #F18* | 48,00 |
| #F19 | 98 | 80 | 96 | #F19* | 48,00 |
| #F20 | 96 | 80 | 96 | #F20* | 44,00 |
| #F21 | 98 | 72 | 96 | #F21* | 44,00 |

**Tabela 7.14:** *Comparativo entre as acurácias (em %) obtidas pelo SVM, MLP, RF e FSTBSVM nas amostras de cebola.*

| Subconjuntos | SVM | MLP | RF | Subconjuntos | FSTBSVM |
|---|---|---|---|---|---|
| #F01 | 94 | 91 | 90 | #F01* | 50,00 |
| #F02 | 92 | 92 | 92 | #F02* | 52,50 |
| #F03 | 97 | 96 | 95 | #F03* | 48,75 |
| #F04 | 99 | 96 | 97 | #F04* | 73,75 |
| #F05 | 99 | 96 | 97 | #F05* | 82,50 |
| #F06 | 99 | 95 | 97 | #F06* | 82,50 |
| #F07 | 97 | 94 | 97 | #F07* | 82,50 |
| #F08 | 97 | 96 | 99 | #F08* | 90,00 |
| #F09 | 99 | 96 | 99 | #F09* | 90,00 |
| #F10 | 100 | 96 | 99 | #F10* | 92,50 |
| #F11 | 100 | 96 | 99 | #F11* | 93,75 |
| #F12 | 100 | 96 | 99 | #F12* | 87,50 |
| #F13 | 100 | 96 | 99 | #F13* | 86,25 |
| #F14 | 99 | 96 | 99 | #F14* | 83,75 |
| #F15 | 99 | 97 | 99 | #F15* | 87,50 |
| #F16 | 97 | 96 | 99 | #F16* | 87,50 |
| #F17 | 99 | 96 | 99 | #F17* | 85,00 |
| #F18 | 99 | 96 | 99 | #F18* | 83,75 |
| #F19 | 99 | 97 | 99 | #F19* | 81,25 |
| #F20 | 99 | 96 | 99 | #F20* | 76,25 |
| #F21 | 97 | 96 | 99 | #F21* | 78,75 |

**Tabela 7.15:** *Comparativo entre as acurácias (em %) obtidas pelo SVM, MLP, RF e FSTBSVM nas amostras de alface.*

| Subconjuntos | SVM | MLP | RF | Subconjuntos | FSTBSVM |
|---|---|---|---|---|---|
| #F01 | 74 | 74 | 59 | #F01* | 52,46 |
| #F02 | 77 | 77 | 69 | #F02* | 57,86 |
| #F03 | 78 | 78 | 67 | #F03* | 66,21 |
| #F04 | 83 | 77 | 78 | #F04* | 69,15 |
| #F05 | 87 | 78 | 86 | #F05* | 75,69 |
| #F06 | 86 | 76 | 84 | #F06* | 73,89 |
| #F07 | 87 | 82 | 83 | #F07* | 79,22 |
| #F08 | 86 | 83 | 85 | #F08* | 82,16 |
| #F09 | 87 | 81 | 86 | #F09* | 83,38 |
| #F10 | 88 | 81 | 86 | #F10* | 85,80 |
| #F11 | 87 | 83 | 90 | #F11* | 86,39 |
| #F12 | 87 | 83 | 89 | #F12* | 88,75 |
| #F13 | 87 | 83 | 89 | #F13* | 86,36 |
| #F14 | 87 | 83 | 90 | #F14* | 85,14 |
| #F15 | 87 | 84 | 89 | #F15* | 85,77 |
| #F16 | 87 | 87 | 89 | #F16* | 86,36 |
| #F17 | 89 | 87 | 89 | #F17* | 86,36 |
| #F18 | 87 | 85 | 89 | #F18* | 86,94 |
| #F19 | 88 | 85 | 89 | #F19* | 84,59 |
| #F20 | 87 | 86 | 89 | #F20* | 83,34 |
| #F21 | 87 | 87 | 88 | #F21* | 82,75 |
| #F22 | 88 | 87 | 89 | #F22* | 83,34 |
| #F23 | 87 | 84 | 89 | #F23* | 83,34 |

Na Tabela (7.16) são apresentados os resultados do SVM combinado com o seletor de variáveis RFI, que obteve 90,50% de acurácia utilizando um conjunto com nove variáveis, enquanto o FSTBSVM, utilizando também um conjunto com nove variáveis, alcançou uma acurácia de 96,00%.

**Tabela 7.16:** *Comparativo entre as acurácias (em %) obtidas pelo SVM e FSTBSVM nas amostras de vinhos produzidos das uvas syrah.*

| Subconjuntos | SVM (RFI) | Subconjuntos | FSTBSVM |
|---|---|---|---|
| #F01 | 85,50 | #F01* | 50,00 |
| #F02 | 85,50 | #F02* | 50,00 |
| #F03 | 90,00 | #F03* | 50,00 |
| #F04 | 88,50 | #F04* | 50,00 |
| #F05 | 88,50 | #F05* | 54,00 |
| #F06 | 86,00 | #F06* | 73,50 |
| #F07 | 86,00 | #F07* | 78,00 |
| #F08 | 88,50 | #F08* | 78,50 |
| #F09 | 90,50 | #F09* | 96,00 |
| #F10 | 85,50 | #F10* | 87,50 |
| #F11 | 83,50 | #F11* | 89,50 |
| #F12 | 83,50 | #F12* | 90,00 |
| #F13 | 85,50 | #F13* | 81,50 |
| #F14 | 80,50 | #F14* | 76,50 |
| #F15 | 79,00 | #F15* | 78,50 |
| #F16 | 78,50 | #F16* | 78,50 |
| #F17 | 80,50 | #F17* | 64,00 |
| #F18 | 71,83 | #F18* | 68,50 |
| #F19 | 74,33 | #F19* | 68,50 |
| #F20 | 76,83 | #F20* | 68,50 |

## 7.7 Considerações Finais

Nesse capítulo, foram apresentados quatro aplicações. A primeira foi realizada na classificação de amostras de laranjas em relação ao sistema de cultivos (orgânico e convencional). Esse estudo resultou em um artigo publicado no *Journal Information Processing in Agriculture* com o título: *Multielement determination in orange juice by ICP-MS associated with data mining for the classification of organic samples* [162]. A segunda aplicação foi em relação à classificação de amostras de vinhos produzidos com uvas Syrah, provenientes das regiões de Mendoza (Argentina) e Vale Central (Chile). Esse estudo culminou em um artigo publicado no *Journal Measurement*, com o título: *Geographical recognition of Syrah wines by combining feature selection with Extreme Learning Machine* [32]. A terceira aplicação utilizou impressão digital (elementos químicos) de alguns vegetais (tomate, alface, cebola e pimentão) combinadas com técnicas de aprendi-

zado de máquina para avaliar a autenticidade de vegetais orgânicos e convencionais. Esse projeto gerou um artigo que está submetido em um *Journal* com o título *Fingerprinting and machine learning techniques to evaluate the authenticity of organic and conventional vegetables*. A quarta aplicação foi em relação a bases de dados médicos disponíveis no repositório da UCI Irvine, e utilizamos o seletor de variáveis FSTBSVM. Esse estudo culminou em um artigo que está submetido em um *Journal* com o título "*Medical data set classification using a new feature selection algorithm combined with twin bounded support vector machine*".

CAPÍTULO 8

# Conclusões e Trabalhos Futuros

Neste trabalho, realizamos um estudo detalhado sobre as SVMs, TWSVM e suas extensões. Apresentamos um algoritmo para problemas de classificação de múltiplas classes, utilizando a estratégia "$1-\times-1-\times-$ *restante*". Nesse algoritmo, abordamos a versão de mínimos quadrados para problemas de classificação com múltiplas classes baseado no TWSVM. Implementamos o princípio de minimização do risco estrutural que auxilia na capacidade de generalização. Nessa linha de estudo do TWSVM, propusemos um seletor de variáveis, para resolver o problema que surge quando se trabalha com grandes conjuntos de dados, onde aparecem muitas variáveis, fazendo com que os algoritmos consumam mais tempo, e em alguns casos, muitas dessas variáveis são redundantes, ou até mesmo irrelevantes, não contribuindo para uma boa classificação. Para validar nosso seletor de variáveis utilizamos bancos de dados médicos abertos e realizamos um estudo comparativo entre os resultados obtidos na classificação de produtos cultivados nos sistemas orgânicos e convencionais e origem geográfica de vinhos, do mesmo modo quando utilizamos os seletores *F-score* e RFI.

Para complementar nosso trabalho, apresentamos aplicações que combinam o uso de ferramentas analíticas associadas a algoritmos de mineração de dados, em banco de dados contendo amostras de alimentos (laranja, alface, cebola, tomate e pimentão) que foram cultivados nos sistemas orgânicos e convencional, com o objetivo de classificá-los quanto ao seu sistema de cultivo. É sabido que os alimentos orgânicos agregam maior valor de mercado, quando comparado a produtos cultivados convencionalmente. Um alimento cultivado convencionalmente poderia ser erroneamente rotulado como orgânico, visto que não há diferenças visuais entre esses produtos. Devido a isso, a autenticação de alimentos é de interesse para muitas comunidades científicas, órgãos reguladores e, principalmente, consumidores.

Ainda quanto à classificação de outros produtos, o interesse em classificar os vinhos com base na variedade das uvas e origem geográfica aumentou nos últimos anos. O solo, as condições climáticas, o tipo de colheita e as condições de produção contribuem para as características que tornam um vinho único. A ideia é garantir que apenas vinhos originados em determinada região sejam permitidos no mercado, protegendo a reputação

da região e assegurando um padrão de qualidade. Juntamente com este conceito, surgiu a necessidade de classificar os vinhos de forma objetiva, a fim de realizar controles de qualidade e evitar fraudes. Nesse viés contribuímos com um estudo em relação ao uso de ferramentas quimiométricas combinadas com algoritmos de mineração de dados, neste caso, na classificação quanto à origem geográfica de vinhos produzidos com uvas do tipo Syrah, produzidos nos países Chile e Argentina.

Por fim, como proposta geral de trabalhos futuros, podemos citar a realização de mais estudos exploratórios envolvendo tanto as SVMs, quanto o TWSVM e suas extensões. Pretendemos estender a versão do algoritmo *support vector classification-regression for K-class* (K-SVCR) para o caso mínimos quadrados e explorar o método de convergência quadrática de Newton com o tamanho do passo de Armijo em versões do algoritmos TWSVM. Outra proposta interessante será propor uma paralelização no algoritmo apresentado nessa tese para resolução de problemas de múltiplas classes.

# Referências Bibliográficas


[1] ABONYI, J.; SZEIFERT, F. **Supervised fuzzy clustering for the identification of fuzzy classifiers**. *Pattern Recognition Letters*, 24(14):2195–2207, 2003.

[2] AKAY, M. F. **Support vector machines combined with feature selection for breast cancer diagnosis**. *Expert systems with applications*, 36(2):3240–3247, 2009.

[3] ALAM, S.; KWON, G.-R.; KIM, J.-I.; PARK, C.-S. **Twin svm-based classification of alzheimer's disease using complex dual-tree wavelet principal coefficients and lda**. *Journal of healthcare engineering*, 2017, 2017.

[4] ANGULO, C.; PARRA, X.; CATALA, A. **K-svcr. a support vector machine for multi-class classification**. *Neurocomputing*, 55(1):57–77, 2003.

[5] ARJUNAN, S.; KUMAR, D.; NAIK, G. **A machine learning based method for classification of fractal features of forearm semg using twin support vector machines**. In: *2010 Annual International Conference of the IEEE Engineering in Medicine and Biology*, p. 4821–4824. IEEE, 2010.

[6] ASCHEMANN-WITZEL, J.; MAROSCHECK, N.; HAMM, U. **Are organic consumers preferring or avoiding foods with nutrition and health claims?** *Food Quality and Preference*, 30(1):68–76, 2013.

[7] BAI, L.; WANG, Z.; SHAO, Y.-H.; DENG, N.-Y. **A novel feature selection method for twin support vector machine**. *Knowledge-Based Systems*, 59:1–8, 2014.

[8] BENNETT, K. P.; BLUE, J. A. **A support vector machine approach to decision trees**. In: *Neural Networks Proceedings, 1998. IEEE World Congress on Computational Intelligence. The 1998 IEEE International Joint Conference on*, volume 3, p. 2396–2401. IEEE, 1998.

[9] BERTSEKAS, D. P. **Nonlinear programming**. Athena scientific Belmont, 1999.

[10] BI, J.; BENNETT, K. P. **A geometric approach to support vector regression**. *Neurocomputing*, 55(1):79–108, 2003.

[11] BIGOT, C.; MEILE, J.-C.; KAPITAN, A.; MONTET, D. **Discriminating organic and conventional foods by analysis of their microbial ecology: An application on fruits**. *Food control*, 48:123–129, 2015.

[12] BLUM, A. L.; LANGLEY, P. **Selection of relevant features and examples in machine learning**. *Artificial intelligence*, 97(1):245–271, 1997.

[13] BOSER, B. E.; GUYON, I. M.; VAPNIK, V. N. **A training algorithm for optimal margin classifiers**. In: *Proceedings of the fifth annual workshop on Computational learning theory*, p. 144–152. ACM, 1992.

[14] BOYD, S.; VANDENBERGHE, L. **Convex optimization**. Cambridge university press, 2004.

[15] BURGES, C. J. **A tutorial on support vector machines for pattern recognition**. *Data mining and knowledge discovery*, 2(2):121–167, 1998.

[16] CAMPBELL, C. **An introduction to kernel methods**. *Studies in Fuzziness and Soft Computing*, 66:155–192, 2001.

[17] CHANDRASHEKAR, G.; SAHIN, F. **A survey on feature selection methods**. *Computers & Electrical Engineering*, 40(1):16–28, 2014.

[18] CHARLEBOIS, S.; SCHWAB, A.; HENN, R.; HUCK, C. W. **Food fraud: An exploratory study for measuring consumer perception towards mislabeled food products and influence on self-authentication intentions**. *Trends in Food Science & Technology*, 50:211–218, 2016.

[19] CHEN, H.-L.; YANG, B.; LIU, J.; LIU, D.-Y. **A support vector machine classifier with rough set-based feature selection for breast cancer diagnosis**. *Expert Systems with Applications*, 38(7):9014–9022, 2011.

[20] CHEN, J.; JI, G. **Multi-class lstsvm classifier based on optimal directed acyclic graph**. In: *Computer and Automation Engineering (ICCAE), 2010 The 2nd International Conference on*, volume 3, p. 100–104. IEEE, 2010.

[21] CHEN, J.; JI, G. **Weighted least squares twin support vector machines for pattern classification**. In: *Computer and Automation Engineering (ICCAE), 2010 The 2nd International Conference on*, volume 2, p. 242–246. IEEE, 2010.

[22] CHEN, S.; XU, J. **Least squares twin support vector machine for multi-class classification**. *International Journal of Database Theory and Application*, 8(5):65–76, 2015.

[23] CHEN, W.-J.; SHAO, Y.-H.; DENG, N.-Y.; FENG, Z.-L. **Laplacian least squares twin support vector machine for semi-supervised classification**. *Neurocomputing*, 145:465–476, 2014.

[24] CHEN, W.-J.; SHAO, Y.-H.; LI, C.-N.; DENG, N.-Y. **Mltsvm: a novel twin support vector machine to multi-label learning**. *Pattern Recognition*, 52:61–74, 2016.

[25] CHEN, X.; YANG, J.; YE, Q.; LIANG, J. **Recursive projection twin support vector machine via within-class variance minimization**. *Pattern Recognition*, 44(10):2643–2655, 2011.

[26] CHEN, Y.-W.; LIN, C.-J. **Combining SVMs with various feature selection strategies**. Springer, 2006.

[27] CONG, H.; YANG, C.; PU, X. **Efficient speaker recognition based on multi-class twin support vector machines and gmms**. In: *2008 IEEE conference on robotics, automation and mechatronics*, 2008.

[28] CORTES, C.; VAPNIK, V. **Support-vector networks**. *Machine learning*, 20(3):273–297, 1995.

[29] COVER, T. M. **Geometrical and statistical properties of systems of linear inequalities with applications in pattern recognition**. *IEEE transactions on electronic computers*, EC-14(3):326–334, 1965.

[30] CRISTIANINI, N.; SHAWE-TAYLOR, J. **An introduction to support vector machines and other kernel-based learning methods**. Cambridge university press, 2000.

[31] DA COSTA, N. L.; CASTRO, I. A.; BARBOSA, R. **Classification of cabernet sauvignon from two different countries in south america by chemical compounds and support vector machines**. *Applied Artificial Intelligence*, 30(7):679–689, 2016.

[32] DA COSTA, N. L.; LLOBODANIN, L. A. G.; LIMA, M. D. D.; CASTRO, I. A.; BARBOSA, R. **Geographical recognition of syrah wines by combining feature selection with extreme learning machine**. *Measurement*, 120:92–99, 2018.

[33] DANEZIS, G. P.; TSAGKARIS, A. S.; BRUSIC, V.; GEORGIOU, C. A. **Food authentication: state of the art and prospects**. *Current Opinion in Food Science*, 10:22–31, 2016.

[34] DE VITIVINICULTURA DA ARGENTINA, I. N. **Informe variedad syrah, departamento de estadística y estudios de mercado**. http://www.inv.gov.ar/inv_

contenidos/pdf/Informe_Variedad_Syrah_-_marzo_2015.pdf, mar 2015. Accessed: 2016.05.20.

[35] DEMŠAR, J. **Statistical comparisons of classifiers over multiple data sets**. *Journal of Machine learning research*, 7(Jan):1–30, 2006.

[36] DHEERU, D.; KARRA TANISKIDOU, E. **UCI machine learning repository**, 2017.

[37] DING, S.; AN, Y.; ZHANG, X.; WU, F.; XUE, Y. **Wavelet twin support vector machines based on glowworm swarm optimization**. *Neurocomputing*, 225:157–163, 2017.

[38] DING, S.; HUA, X. **Recursive least squares projection twin support vector machines for nonlinear classification**. *Neurocomputing*, 130:3–9, 2014.

[39] DING, S.; ZHAO, X.; ZHANG, J.; ZHANG, X.; XUE, Y. **A review on multi-class twsvm**. *Artificial Intelligence Review*, p. 1–27, 2017.

[40] DING, X.; ZHANG, G.; KE, Y.; MA, B.; LI, Z. **High efficient intrusion detection methodology with twin support vector machines**. In: *2008 International Symposium on Information Science and Engineering*, volume 1, p. 560–564. IEEE, 2008.

[41] EVGENIOU, T.; PONTIL, M.; POGGIO, T. **Regularization networks and support vector machines**. *Advances in computational mathematics*, 13(1):1–50, 2000.

[42] FUNG, G.; MANGASARIAN, O. L.; SHAVLIK, J. W. **Knowledge-based support vector machine classifiers**. In: *NIPS*, p. 521–528, 2002.

[43] FUNG, G. M.; MANGASARIAN, O. L. **Proximal support vector machine classifiers**. *Proceedings KDD-2001: Knowledge Discovery and Data Mining, F. Provost and R. Srikant, editors,*, p. 77–86, 2001.

[44] GAO, S.; YE, Q.; YE, N. **1-norm least squares twin support vector machines**. *Neurocomputing*, 74(17):3590–3597, 2011.

[45] GHORAI, S.; HOSSAIN, S. J.; MUKHERJEE, A.; DUTTA, P. K. **Newton's method for nonparallel plane proximal classifier with unity norm hyperplanes**. *Signal Processing*, 90(1):93–104, 2010.

[46] GHORAI, S.; MUKHERJEE, A.; DUTTA, P. K. **Nonparallel plane proximal classifier**. *Signal Processing*, 89(4):510–522, 2009.

[47] GOLUB, G. H.; VAN LOAN, C. F. **Matrix computations. 1996**. *Johns Hopkins University, Press, Baltimore, MD, USA*, p. 374–426, 1996.

[48] GOODMAN, D. E.; BOGGESS, L.; WATKINS, A. **Artificial immune system classification of multiple-class problems**. *Proceedings of the artificial neural networks in engineering ANNIE*, 2:179–183, 2002.

[49] GU, Z.; ZHANG, Z.; SUN, J.; LI, B. **Robust image recognition by l1-norm twin-projection support vector machine**. *Neurocomputing*, 223:1–11, 2017.

[50] GUO, H.; NANDI, A. K. **Breast cancer diagnosis using genetic programming generated feature**. *Pattern Recognition*, 39(5):980–987, 2006.

[51] GUO, J.; YI, P.; WANG, R.; YE, Q.; ZHAO, C. **Feature selection for least squares projection twin support vector machine**. *Neurocomputing*, 144:174–183, 2014.

[52] GUO, Y.; DONG, M.; YANG, Z.; GAO, X.; WANG, K.; LUO, C.; MA, Y.; ZHANG, J. **A new method of detecting micro-calcification clusters in mammograms using contourlet transform and non-linking simplified pcnn**. *Computer methods and programs in biomedicine*, 130:31–45, 2016.

[53] GUYON, I.; WESTON, J.; BARNHILL, S.; VAPNIK, V. **Gene selection for cancer classification using support vector machines**. *Machine learning*, 46(1-3):389–422, 2002.

[54] HAMILTON, H. J.; SHAN, N.; CERCONE, N. **RIAC: a rule induction algorithm based on approximate classification**. Computer Science Department, University of Regina, 1996.

[55] HAO, P.-Y. **New support vector algorithms with parametric insensitive/margin model**. *Neural Networks*, 23(1):60–73, 2010.

[56] HAYKIN, S. S. **Neural networks and learning machines**, volume 3. Pearson Upper Saddle River, NJ, USA:, 2009.

[57] HE, J.; ZHENG, S.-H. **Intrusion detection model with twin support vector machines**. *Journal of Shanghai Jiaotong University (Science)*, 19:448–454, 2014.

[58] HEASRT, M.; SCHOLKOPF, B.; DUMAIS, S.; OSUNA, E.; PLATT, J. **Trends and controversies-support vector machines**. *IEEE Intelligent Systems*, 13(4):18–28, 1998.

[59] HOLM, S. **A simple sequentially rejective multiple test procedure**. *Scandinavian journal of statistics*, p. 65–70, 1979.

[60] HOU, Q.; LIU, L.; ZHEN, L.; JING, L. **A novel projection nonparallel support vector machine for pattern classification**. *Engineering Applications of Artificial Intelligence*, 75:64–75, 2018.

[61] HSU, C.-W.; LIN, C.-J. **A comparison of methods for multiclass support vector machines**. *IEEE transactions on Neural Networks*, 13(2):415–425, 2002.

[62] HUA, X.; DING, S. **Weighted least squares projection twin support vector machines with local information**. *Neurocomputing*, 160:228–237, 2015.

[63] HUANG, H.-J.; DING, S.-F.; SHI, Z.-Z. **Primal least squares twin support vector regression**. *Journal of Zhejiang University SCIENCE C*, 14(9):722–732, 2013.

[64] HUANG, H.; WEI, X.; ZHOU, Y. **A sparse method for least squares twin support vector regression**. *Neurocomputing*, 211:150–158, 2016.

[65] INSTITUTE, W. **Wold statistics, world wine consumption by country**. http://www.wineinstitute.org/resources/statistics, 2014. Accessed: 2016.05.20.

[66] JAYADEVA, KHEMCHANDANI, R.; CHANDRA, S. **Twin support vector machines for pattern classification**. *IEEE Transactions on pattern analysis and machine intelligence*, 29(5):905–910, 2007.

[67] JAYADEVA, R. K.; CHANDRA, S.; OTHERS. **Twin support vector machines: Models, extensions and applications**, 2016.

[68] JURADO, J. M.; ALCÁZAR, Á.; PALACIOS-MORILLO, A.; DE PABLOS, F. **Classification of spanish do white wines according to their elemental profile by means of support vector machines**. *Food chemistry*, 135(3):898–903, 2012.

[69] KHEMCHANDANI, RESHMA, J.; CHANDRA, S. **Knowledge based proximal support vector machines**. *European Journal of Operational Research*, 195(3):914–923, 2009.

[70] KHEMCHANDANI, R.; CHANDRA, S. **Fuzzy twin support vector machines for pattern classification**. *Mathematical Programming and Game Theory for Decision Making*, 1:131, 2008.

[71] KHEMCHANDANI, R.; GOYAL, K.; CHANDRA, S. **Twsvr: regression via twin support vector machine**. *Neural Networks*, 74:14–21, 2016.

[72] KHEMCHANDANI, R.; PAL, A.; CHANDRA, S. **Fuzzy least squares twin support vector clustering**. *Neural computing and applications*, 29(2):553–563, 2018.

[73] KHEMCHANDANI, R.; SAIGAL, P.; CHANDRA, S. **Improvements on ν-twin support vector machine**. *Neural Networks*, 79:97–107, 2016.

[74] KHEMCHANDANI, R.; SHARMA, S. **Robust least squares twin support vector machine for human activity recognition**. *Applied Soft Computing*, 47:33–46, 2016.

[75] KOHAVI, R.; OTHERS. **A study of cross-validation and bootstrap for accuracy estimation and model selection**. In: *Ijcai*, volume 14, p. 1137–1145. Montreal, Canada, 1995.

[76] KUMAR, D.; THAKUR, M. **Weighted multicategory nonparallel planes svm classifiers**. *Neurocomputing*, 211:106–116, 2016.

[77] KUMAR, D.; THAKUR, M. **All-in-one multicategory least squares nonparallel hyperplanes support vector machine**. *Pattern Recognition Letters*, 105:165–174, 2018.

[78] KUMAR, M. A.; GOPAL, M. **Application of smoothing technique on twin support vector machines**. *Pattern Recognition Letters*, 29(13):1842–1848, 2008.

[79] KUMAR, M. A.; GOPAL, M. **Least squares twin support vector machines for pattern classification**. *Expert Systems with Applications*, 36(4):7535–7543, 2009.

[80] KUMAR, M. A.; KHEMCHANDANI, R.; GOPAL, M.; CHANDRA, S. **Knowledge based least squares twin support vector machines**. *Information Sciences*, 180(23):4606–4618, 2010.

[81] LANTZ, B. **Machine learning with R**. Packt Publishing Ltd, 2013.

[82] LAURSEN, K.; MIHAILOVA, A.; KELLY, S.; EPOV, V.; BÉRAIL, S.; SCHJOERRING, J.; DONARD, O.; LARSEN, E. H.; PEDENTCHOUK, N.; MARCA-BELL, A.; OTHERS. **Is it really organic?–multi-isotopic analysis as a tool to discriminate between organic and conventional plants**. *Food chemistry*, 141(3):2812–2820, 2013.

[83] LEE, M.-C. **Using support vector machine with a hybrid feature selection method to the stock trend prediction**. *Expert Systems with Applications*, 36(8):10896–10904, 2009.

[84] LEE, Y.-J.; MANGASARIAN, O. L. **Ssvm: A smooth support vector machine for classification**. *Computational optimization and Applications*, 20(1):5–22, 2001.

[85] LI, D.; TIAN, Y. **Twin support vector machine in linear programs**. *Procedia Computer Science*, 29:1770–1778, 2014.

[86] LI, J.-B.; PENG, Y.; LIU, D. **Quasiconformal kernel common locality discriminant analysis with application to breast cancer diagnosis**. *Information Sciences*, 223:256–269, 2013.

[87] LICHMAN, M. **Uci machine learning repository. irvine, ca: University of california, school of information and computer science**. *URL http://archive. ics. uci. edu/ml*, 2013.

[88] LIMA, E. L. **Espaços métricos**, volume 4. Instituto de Matemática Pura e Aplicada, 1983.

[89] LIMA, E. L. **Curso de análise, volumes 1 e 2**. *Projeto Euclides-IMPA, Rio de Janeiro*, 1989.

[90] LIMA, M. D. D.; DA COSTA, N. L.; BARBOSA, R. **Improvements on least squares twin multi-class classification support vector machine**. *Neurocomputing*, 313:196–205, 2018.

[91] LIU, M.; WANG, M.; WANG, J.; LI, D. **Comparison of random forest, support vector machine and back propagation neural network for electronic tongue data classification: Application to the recognition of orange beverage and chinese vinegar**. *Sensors and Actuators B: Chemical*, 177:970–980, 2013.

[92] LIU, M.; XIE, Y.; YAO, Z.; DAI, B. **A new hybrid gmm/svm for speaker verification**. In: *18th International Conference on Pattern Recognition (ICPR'06)*, volume 4, p. 314–317. IEEE, 2006.

[93] LLOBODANIN, L.; BARROSO, L.; CASTRO, I. **Prediction of the functionality of young south american red wines based on chemical parameters**. *Australian journal of grape and wine research*, 20(1):15–24, 2014.

[94] LORENA, A. C.; DE CARVALHO, A. C. **Introdução as máquinas de vetores suporte**. *Relatório Técnico do Instituto de Ciências Matemáticas e de Computaçao (USP/Sao Carlos)*, 192, 2003.

[95] LORENA, A. C.; DE CARVALHO, A. C. **Uma introdução às support vector machines**. *Revista de Informática Teórica e Aplicada*, 14(2):43–67, 2007.

[96] MAIONE, C.; BATISTA, B. L.; CAMPIGLIA, A. D.; BARBOSA, F.; BARBOSA, R. M. **Classification of geographic origin of rice by data mining and inductively coupled plasma mass spectrometry**. *Computers and Electronics in Agriculture*, 121:101–107, 2016.

[97] MALDONADO, S.; LÓPEZ, J. **Synchronized feature selection for support vector machines with twin hyperplanes**. *Knowledge-Based Systems*, 2017.

[98] MANGASARIAN, O. L. **Nonlinear programming**, volume 10. siam, 1993.

[99] MANGASARIAN, O. L.; MUSICANT, D. R. **Successive overrelaxation for support vector machines**. *IEEE Transactions on Neural Networks*, 10(5):1032–1037, 1999.

[100] MANGASARIAN, O. L.; MUSICANT, D. R. **Active support vector machine classification**. In: *NIPS*, p. 577–583, 2000.

[101] MANGASARIAN, O. L.; WILD, E. W. **Multisurface proximal support vector machine classification via generalized eigenvalues**. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(1):69–74, 2006.

[102] MARCONDES FILHO, D. **Cartas de controle multivariadas baseadas no método kernel-statis para monitoramento de processos em bateladas**, 2009.

[103] MERCER, J. **Functions of positive and negative type, and their connection with the theory of integral equations**. *Philosophical transactions of the royal society of London. Series A, containing papers of a mathematical or physical character*, 209:415–446, 1909.

[104] MINISTÉRIO DA AGRICULTURA, P. E. A. **Decreto n. 6.323, de 27 de dezembro de 2007: Dispõe sobre a agricultura orgânica e dá outras providências**. http://www.planalto.gov.br/ccivil_03/_ato2007-2010/2007/decreto/d6323.htm, dez 2007. Acessado: 20/09/2018.

[105] MITCHELL, T. M.; OTHERS. **Machine learning. wcb**, 1997.

[106] MOZAFARI, K.; NASIRI, J. A.; CHARKARI, N. M.; JALILI, S. **Action recognition by local space-time features and least square twin svm (ls-tsvm)**. In: *Informatics and Computational Intelligence (ICI), 2011 First International Conference on*, p. 287–292. IEEE, 2011.

[107] MUNOZ, A. **Machine learning and optimization**. https://www.cims.nyu.edu/~{}munoz/files/ml_optimization.pdf, 2014. Acessado: 20/09/2018.

[108] NAIK, G. R.; KUMAR, D. K.; OTHERS. **Twin svm for gesture classification using the surface electromyogram**. *IEEE Transactions on Information Technology in Biomedicine*, 14(2):301–308, 2010.

[109] NASIRI, J. A.; CHARKARI, N. M.; JALILI, S. **Least squares twin multi-class classification support vector machine**. *Pattern Recognition*, 48(3):984–992, 2015.

[110] NASIRI, J. A.; CHARKARI, N. M.; MOZAFARI, K. **Energy-based model of least squares twin support vector machines for human action recognition**. *Signal Processing*, 104:248–257, 2014.

[111] PAN, X.; LUO, Y.; XU, Y. **K-nearest neighbor based structural twin support vector machine**. *Knowledge-Based Systems*, 88:34–44, 2015.

[112] PARASTALOOI, N.; AMIRI, A.; ALIHEIDARI, P. **Modified twin support vector regression**. *Neurocomputing*, 211:84–97, 2016.

[113] PARLETT, B. N. **The symmetric eigenvalue problem**, volume 7. SIAM, 1980.

[114] PENA-REYES, C. A.; SIPPER, M. **A fuzzy-genetic approach to breast cancer diagnosis**. *Artificial intelligence in medicine*, 17(2):131–155, 1999.

[115] PENG, L.; CHEN, W.; ZHOU, W.; LI, F.; YANG, J.; ZHANG, J. **An immune-inspired semi-supervised algorithm for breast cancer diagnosis**. *Computer methods and programs in biomedicine*, 134:259–265, 2016.

[116] PENG, X. **A ν-twin support vector machine (ν-tsvm) classifier and its geometric algorithms**. *Information Sciences*, 180(20):3863–3875, 2010.

[117] PENG, X. **Tsvr: an efficient twin support vector machine for regression**. *Neural Networks*, 23(3):365–372, 2010.

[118] PENG, X. **Tpmsvm: a novel twin parametric-margin support vector machine for pattern recognition**. *Pattern Recognition*, 44(10):2678–2692, 2011.

[119] PENG, X.; KONG, L.; CHEN, D. **Improvements on twin parametric-margin support vector machine**. *Neurocomputing*, 151:857–863, 2015.

[120] PENG, X.; XU, D. **Bi-density twin support vector machines for pattern recognition**. *Neurocomputing*, 99:134–143, 2013.

[121] PENG, X.; XU, D. **Structural regularized projection twin support vector machine for data classification**. *Information Sciences*, 279:416–432, 2014.

[122] PENG, X.; XU, D.; SHEN, J. **A twin projection support vector machine for data regression**. *Neurocomputing*, 138:131–141, 2014.

[123] PERIS, M.; ESCUDER-GILABERT, L. **Electronic noses and tongues to assess food authenticity and adulteration**. *Trends in Food Science & Technology*, 58:40–54, 2016.

[124] PLATT, J. C.; CRISTIANINI, N.; SHAWE-TAYLOR, J. **Large margin dags for multiclass classification**. In: *Proceedings of the 12th International Conference on Neural Information Processing Systems*, p. 547–553. MIT press, 1999.

[125] POLAT, K.; GÜNEŞ, S. **Breast cancer diagnosis using least square support vector machine**. *Digital Signal Processing*, 17(4):694–701, 2007.

[126] PROVOST, F. **Glossary of terms special issue on applications of machine learning and the knowledge discovery process**, 1998.

[127] QI, Z.; TIAN, Y.; SHI, Y. **Robust twin support vector machine for pattern classification**. *Pattern Recognition*, 46(1):305–316, 2013.

[128] QI, Z.; TIAN, Y.; SHI, Y. **Structural twin support vector machine for classification**. *Knowledge-Based Systems*, 43:74–81, 2013.

[129] QI, Z.; TIAN, Y.; SHI, Y. **A nonparallel support vector machine for a classification problem with universum learning**. *journal of Computational and Applied Mathematics*, 263:288–298, 2014.

[130] QUINLAN, J. R. **Improved use of continuous attributes in c4. 5**. *Journal of artificial intelligence research*, 4:77–90, 1996.

[131] RASTOGI, R.; SAIGAL, P.; CHANDRA, S. **Angle-based twin parametric-margin support vector machine for pattern classification**. *Knowledge-Based Systems*, 139:64–77, 2018.

[132] RICHHARIYA, B.; TANVEER, M. **A robust fuzzy least squares twin support vector machine for class imbalance learning**. *Applied Soft Computing*, 71:418–432, 2018.

[133] RICHHARIYA, B.; TANVEER, M. **Eeg signal classification using universum support vector machine**. *Expert Systems with Applications*, 106:169–182, 2018.

[134] RUSSELL, S.; NORVIG, P.; INTELLIGENCE, A. **A modern approach**. *Artificial Intelligence. Prentice-Hall, Egnlewood Cliffs*, 25, 1995.

[135] SAG. **Servicio agricola y ganadero, informe ejecutivo: Existencia de vinos al 31/12/2015**. http://www.sag.gob.cl/sites/default/files/informe_de_extistencias_2015.pdf, dec 2015. Accessed: 2016.05.20.

[136] SCHÖLKOPF, B.; SMOLA, A. J.; WILLIAMSON, R. C.; BARTLETT, P. L. **New support vector algorithms**. *Neural computation*, 12(5):1207–1245, 2000.

[137] SCHÖLKOPF, B.; SMOLA, A. J. **Learning with kernels: support vector machines, regularization, optimization, and beyond**. MIT press, 2002.

[138] SEJDINOVIC, D.; GRETTON, A. **What is an RKHS?** Citeseer, 2012.

[139] SHAO, Y.-H.; CHEN, W.-J.; DENG, N.-Y. **Nonparallel hyperplane support vector machine for binary classification problems**. *Information Sciences*, 263:22–35, 2014.

[140] SHAO, Y.-H.; CHEN, W.-J.; HUANG, W.-B.; YANG, Z.-M.; DENG, N.-Y. **The best separating decision tree twin support vector machine for multi-class classification**. *Procedia Computer Science*, 17:1032–1038, 2013.

[141] SHAO, Y.-H.; CHEN, W.-J.; WANG, Z.; LI, C.-N.; DENG, N.-Y. **Weighted linear loss twin support vector machine for large-scale classification**. *Knowledge-Based Systems*, 73:276–288, 2015.

[142] SHAO, Y.-H.; CHEN, W.-J.; ZHANG, J.-J.; WANG, Z.; DENG, N.-Y. **An efficient weighted lagrangian twin support vector machine for imbalanced data classification**. *Pattern Recognition*, 47(9):3158–3167, 2014.

[143] SHAO, Y.-H.; DENG, N.-Y. **A novel margin-based twin support vector machine with unity norm hyperplanes**. *Neural Computing and Applications*, 22(7-8):1627–1635, 2013.

[144] SHAO, Y.-H.; DENG, N.-Y.; YANG, Z.-M. **Least squares recursive projection twin support vector machine for classification**. *Pattern Recognition*, 45(6):2299–2307, 2012.

[145] SHAO, Y.-H.; WANG, Z.; CHEN, W.-J.; DENG, N.-Y. **A regularization for the projection twin support vector machine**. *Knowledge-Based Systems*, 37:203–210, 2013.

[146] SHAO, Y.-H.; ZHANG, C.-H.; WANG, X.-B.; DENG, N.-Y. **Improvements on twin support vector machines**. *Neural Networks, IEEE Transactions on*, 22(6):962–968, 2011.

[147] SMOLA, A. J.; SCHÖLKOPF, B. **A tutorial on support vector regression**. *Statistics and computing*, 14(3):199–222, 2004.

[148] SONGSIRI, P.; PHETKAEW, T.; KIJSIRIKUL, B. **Enhancement of multi-class support vector machine construction from binary learners using generalization performance**. *Neurocomputing*, 151:434–448, 2015.

[149] TANG, Y.; ZHANG, Y.-Q.; CHAWLA, N. V.; KRASSER, S. **Svms modeling for highly imbalanced classification**. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 39(1):281–288, 2009.

[150] TANVEER, M.; KHAN, M. A.; HO, S.-S. **Robust energy-based least squares twin support vector machines**. *Applied Intelligence*, 45(1):174–186, 2016.

[151] TAROZZI, A.; HRELIA, S.; ANGELONI, C.; MORRONI, F.; BIAGI, P.; GUARDIGLI, M.; CANTELLI-FORTI, G.; HRELIA, P. **Antioxidant effectiveness of organically and non-organically grown red oranges in cell culture systems**. *European Journal of Nutrition*, 45(3):152–158, 2006.

[152] TIAN, Y.; JU, X.; QI, Z.; SHI, Y. **Improved twin support vector machine**. *Science China Mathematics*, 57(2):417–432, 2014.

[153] TIKHONOV, A. N.; ARSENIN, V. Y. **Solutions of ill-posed problems**. Winston, 1977.

[154] TOMAR, D.; AGARWAL, S. **Feature selection based least square twin support vector machine for diagnosis of heart disease**. *International Journal of Bio-Science and Bio-Technology*, 6(2):69–82, 2014.

[155] TOMAR, D.; AGARWAL, S. **Predictive model for diabetic patients using hybrid twin support vector machine**. In: *Proceedings of the 5th International Conferences on Advances in Communication Network and Computing (CNC'14)*, p. 1–9, 2014.

[156] TOMAR, D.; AGARWAL, S. **A comparison on multi-class classification methods based on least squares twin support vector machine**. *Knowledge-Based Systems*, 81:131–147, 2015.

[157] TOMAR, D.; AGARWAL, S. **Twin support vector machine: a review from 2007 to 2014**. *Egyptian Informatics Journal*, 16(1):55–69, 2015.

[158] TOMAR, D.; AGARWAL, S. **Multi-class twin support vector machine for pattern classification**. In: *Proceedings of 3rd International Conference on Advanced Computing, Networking and Informatics*, p. 97–110. Springer, 2016.

[159] TOMAR, D.; OJHA, D.; AGARWAL, S. **An emotion detection system based on multi least squares twin support vector machine**. *Advances in Artificial Intelligence*, 2014:8, 2014.

[160] TOMAR, D.; SINGHAL, S.; AGARWAL, S. **Weighted least square twin support vector machine for imbalanced dataset**. *International Journal of Database Theory and Application*, 7(2):25–36, 2014.

[161] TURRA, C.; FERNANDES, E.; BACCHI, M.; TAGLIAFERRO, F.; FRANÇA, E. **Differences between elemental composition of orange juices and leaves from organic**

**and conventional production systems**. *Journal of radioanalytical and nuclear chemistry*, 270(1):203–208, 2006.

[162] TURRA, C.; DE LIMA, M. D.; FERNANDES, E. A. D. N.; BACCHI, M. A.; BARBOSA, F.; BARBOSA, R. **Multielement determination in orange juice by icp-ms associated with data mining for the classification of organic samples**. *Information Processing in Agriculture*, 4:199–205, 2017.

[163] VAPNIK, V. **The nature of statistical learning theory**. *New York: Springer*, 1995.

[164] VAPNIK, V. N. **An overview of statistical learning theory**. *IEEE transactions on neural networks*, 10(5):988–999, 1999.

[165] VAPNIK, V. N.; VAPNIK, V. **Statistical learning theory**, volume 1. Wiley New York, 1998.

[166] WANG, B. X.; JAPKOWICZ, N. **Boosting support vector machines for imbalanced data sets**. *Knowledge and information systems*, 25(1):1–20, 2010.

[167] WANG, D.; YE, N.; YE, Q. **Twin support vector machines via fast generalized newton refinement**. In: *Intelligent Human-Machine Systems and Cybernetics (IHMSC), 2010 2nd International Conference on*, volume 2, p. 62–65. IEEE, 2010.

[168] WANG, H.; ZHOU, Z.; XU, Y. **An improved $\nu$-twin bounded support vector machine**. *Applied Intelligence*, p. 1–13, 2017.

[169] WANG, Z.; SHAO, Y.-H.; WU, T.-R. **A ga-based model selection for smooth twin parametric-margin support vector machine**. *Pattern Recognition*, 46(8):2267–2277, 2013.

[170] WESTON, J.; COLLOBERT, R.; SINZ, F.; BOTTOU, L.; VAPNIK, V. **Inference with the universum**. In: *Proceedings of the 23rd international conference on Machine learning*, p. 1009–1016. ACM, 2006.

[171] WU, X.; KUMAR, V.; QUINLAN, J. R.; GHOSH, J.; YANG, Q.; MOTODA, H.; MCLACHLAN, G. J.; NG, A.; LIU, B.; PHILIP, S. Y.; OTHERS. **Top 10 algorithms in data mining**. *Knowledge and information systems*, 14(1):1–37, 2008.

[172] XIE, J.; HONE, K.; XIE, W.; GAO, X.; SHI, Y.; LIU, X. **Extending twin support vector machine classifier for multi-category classification problems**. *Intelligent Data Analysis*, 17(4):649–664, 2013.

[173] XIE, X.; SUN, S. **Multitask centroid twin support vector machines**. *Neurocomputing*, 149:1085–1091, 2015.

[174] XU, X. **Generalization of the sherman–morrison–woodbury formula involving the schur complement**. *Applied Mathematics and Computation*, 309:183–191, 2017.

[175] XU, Y. **K-nearest neighbor-based weighted multi-class twin support vector machine**. *Neurocomputing*, 205:430–438, 2016.

[176] XU, Y.; GUO, R. **An improved ν-twin support vector machine**. *Applied intelligence*, 41(1):42–54, 2014.

[177] XU, Y.; GUO, R.; WANG, L. **A twin multi-class classification support vector machine**. *Cognitive computation*, 5(4):580–588, 2013.

[178] XU, Y.; WANG, L.; ZHONG, P. **A rough margin-based ν-twin support vector machine**. *Neural Computing and Applications*, 21(6):1307–1317, 2012.

[179] XU, Y.; XI, W.; LV, X.; GUO, R. **An improved least squares twin support vector machine**. *JOURNAL OF INFORMATION &COMPUTATIONAL SCIENCE*, 9(4):1063–1071, 2012.

[180] XU, Y.; YU, J.; ZHANG, Y. **Knn-based weighted rough ν-twin support vector machine**. *Knowledge-Based Systems*, 71:303–313, 2014.

[181] YANG, C.; JI, L.; LIU, G. **Study to speech emotion recognition based on twinssvm**. In: *2009 Fifth international conference on natural computation*, volume 2, p. 312–316. IEEE, 2009.

[182] YANG, Z.-M.; HE, J.-Y.; SHAO, Y.-H. **Feature selection based on linear twin support vector machines**. *Procedia Computer Science*, 17:1039–1046, 2013.

[183] YANG, Z.-X.; SHAO, Y.-H.; ZHANG, X.-S. **Multiple birth support vector machine for multi-class classification**. *Neural Computing and Applications*, 22(1):153–161, 2013.

[184] YANG, Z.; XU, Y. **Laplacian twin parametric-margin support vector machine for semi-supervised classification**. *Neurocomputing*, 171:325–334, 2016.

[185] YE, Q.; ZHAO, C.; GAO, S.; ZHENG, H. **Weighted twin support vector machines with local information and its application**. *Neural Networks*, 35:31–39, 2012.

[186] YE, Q.; ZHAO, C.; YE, N.; CHEN, Y. **Multi-weight vector projection support vector machines**. *Pattern Recognition Letters*, 31(13):2006–2011, 2010.

[187] ZHANG, X.; DING, S.; XUE, Y. **An improved multiple birth support vector machine for pattern classification**. *Neurocomputing*, 225:119–128, 2017.

[188] ZHANG, X. **Boosting twin support vector machine approach for mcs detection**. In: *Information Processing, 2009. APCIP 2009. Asia-Pacific Conference on*, volume 1, p. 149–152. IEEE, 2009.

[189] ZHANG, X.; GAO, X. **Twin support vector machines and subspace learning methods for microcalcification clusters detection**. *Engineering Applications of Artificial Intelligence*, 25(5):1062–1072, 2012.

[190] ZHANG, X.; GAO, X.; WANG, M. **Mcs detection approach using bagging and boosting based twin support vector machine**. In: *Systems, Man and Cybernetics, 2009. SMC 2009. IEEE International Conference on*, p. 5000–5505. IEEE, 2009.

[191] ZHANG, Z.; ZHEN, L.; DENG, N.; TAN, J. **Sparse least square twin support vector machine with adaptive norm**. *Applied intelligence*, 41(4):1097–1107, 2014.

[192] ZHAO, Y.-P.; ZHAO, J.; ZHAO, M. **Twin least squares support vector regression**. *Neurocomputing*, 118:225–236, 2013.

[193] ZHENG, B.; YOON, S. W.; LAM, S. S. **Breast cancer diagnosis based on feature extraction using a hybrid of k-means and support vector machine algorithms**. *Expert Systems with Applications*, 41(4):1476–1482, 2014.

[194] ZHOU, Z.-H.; LIU, X.-Y. **Training cost-sensitive neural networks with methods addressing the class imbalance problem**. *IEEE Transactions on Knowledge and Data Engineering*, 18(1):63–77, 2006.

[195] ZHU, J.; ROSSET, S.; HASTIE, T.; TIBSHIRANI, R. **1-norm support vector machines**. *Advances in neural information processing systems*, 16(1):49–56, 2004.